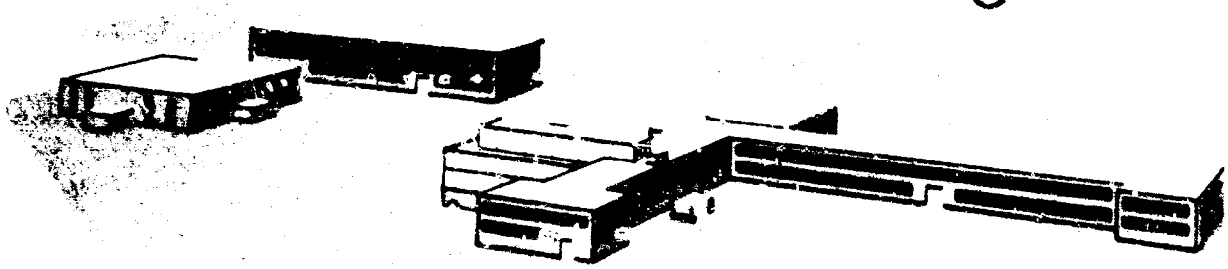
352.14-R-2

A GENERAL MODEL FOR SIMULATING INFORMATION STORAGE AND RETRIEVAL SYSTEMS



H R B - S I N G E R, I N C.
SCIENCE PARK · STATE COLLEGE PENNSYLVANIA

SINGER

"The following publication designations and their descriptions outline the system used by HRB-Singer, Inc., to differentiate the various types of technical publications."

Project No.		Serial No.	
37	-	42	No letter designation is given to Interim Progress Reports.
37 - R	-	42	"R" is designated to all technical reports completing a task on a current project.
37	-	F	"F" is assigned to all Final Report designations.
53 - M	-	46	"M" designation is assigned to all equipment operation and maintenance manuals.
P	-	612	"P" is assigned to all proposals.
S	-	24	"S" describes Special Publications.

HRB-SINGER, INC.

A SUBSIDIARY OF THE SINGER COMPANY

Science Park, State College, Pa.

352.14-R-2

A GENERAL MODEL FOR SIMULATING INFORMATION STORAGE AND RETRIEVAL SYSTEMS

Contract Nonr. 3818(00)
Information Systems Branch
Office of Naval Research

April 1966

Copy No. 2 of 106 Copies

Prepared by:
Charles R. Blunt
Robert T. Duquet
Peter T. Luckie

Contributors:
Priscilla M. Gould
H. Dean Wilson

Reviewed by:
Kenneth F. Barber
Methodology Branch

Approved by:
Ralph C. Stevenson
Behavioristics Laboratory

ABSTRACT

This report presents the results of a research effort to explore the use of computer simulation as a quantitative tool for planning, analyzing and evaluating Information Retrieval (IR) systems. A general time-flow model has been developed that enables a systems engineer to simulate the interactions among personnel, equipment and data at each step in an information processing effort. The input parameters for the simulation reflect the configuration of the system, the processing load of the system, the work schedule of the system, the work schedule of the personnel, equipment availability, the likelihood and effect of errors in processing and the location and availability of the system user. Simulation output provides a study of system response time (both delay time and processing time), equipment and personnel work and idle time and the location and size of data queues.

Included within this report is a discussion of the simulation rationale, the modeling methodology employed and the input and output data of the computer simulation programs. Additionally, one example of a system simulation is presented as an illustration of the capability of this kind of tool in systems analysis.

FOREWORD

A systems engineer, identifying and illustrating the need for an information system, asks --

Will this configuration create an unacceptable level of delay in processing?

What are the overall advantageous of adding a second satellite computer?

A facilities manager, assessing the effectiveness of his information system, asks --

Is there any advantage in rescheduling the availability of the C. P. U. to increase response time?

Are there any unique indicators to warn of an approaching temporary saturation point of the system?

An administrator, evaluating alternative or additional information systems, asks --

At what point can I expect to have to increase the capacity of the system assuming a growth rate of X load per year?

What components of the system must be replaced or expanded to insure continual 100% operation?

This report summarizes the examination of a design and planning model that could be used as a tool to answer these questions. The research was performed to provide methods of evaluating intelligence systems, but the general nature of the model also makes it applicable to other types of Information Systems.

The report is organized into two sections. The main text discusses the concepts of the simulation model and its application. The appendices contain a discussion of the program (including general logic diagrams), the preparation of input data, and an example of output data.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

I. INTRODUCTION

This work is part of a research program sponsored by the Information Systems Branch of the Office of Naval Research under contract Nonr 3818(00) to formulate general purpose simulation models of the various functional components found within intelligence systems. The material presented describes a generalized time-flow model that can be applied in the planning, design and evaluation phases of information storage and retrieval systems.[1] This research report summarizes attempts to produce a general information systems model. It is not the intention of this report to present the model as a final developed simulator, but rather as a base for subsequent development of such an evaluation tool. Some specific aspects to be considered in such a development are set forth at the end of this report.

A. OBJECTIVE

Information storage and retrieval concepts are continuously being proposed as feasible solutions to some of the problems of timely intelligence processing, analysis and dissemination. Experience has shown, however, that there has often been a long costly interval between a design concept and a successfully operating system. At present, one of the more successful (although expensive) methods of testing the feasibility of a concept is to build a pilot configuration for operational experimentation. In this manner, representative problem areas are probed and the findings serve as feedback to the continuing testing and development effort.

Computer simulation of a retrieval system can provide the design engineers with more timely information (at less cost) than is now available from operational experimentation. Therefore, the prime objective of this research effort has been to investigate quantitative aspects of information retrieval systems; in particular, to develop a general model that could yield a measure

[1] The simulation philosophy and initial model development have been detailed in previous reports -- namely, HRB-Singer Report 352-R-17, "The Simulation and Evaluation of Information Retrieval Systems," April 1965 (AD464619) and HRB-Singer Report 352.14-R-1, "An Information Retrieval System Model," October 1965 (AD623590).

of a system's performance. Such a model should be adaptable to any specific system; i.e., to different mixes of equipment, personnel, and procedures.

B. RATIONALE

In the development of an information storage and retrieval system, certain basic elements directly relate to satisfying the system user's requirements; namely, quality of presentation, cost of operation, and system response time. These elements can be considered measurements of the "energy" necessary to produce the desired output from the total file.

1. Quality of Presentation

An effective system should be sensitive to a user's information needs. A request should be answered with a complete output of relevant information within the desired time. If this quality is defined as system effectiveness, then effectiveness is a judgment and is a difficult measure of an intelligence system's performance. The data within an intelligence system are often incomplete, inaccurate and sometimes invalid. The significance of a single item may often outweigh the utility of hundreds of reports.

2. Cost of Operation

The operating cost of an information system is the sum of the operating costs of each function (e.g., data collection, input preparation, storage, retrieval and presentation) plus the maintenance and support costs incurred to maintain the operations. Sometimes the operating costs may also include initial costs prorated over several years. Initial costs may include expenditures for research, development, equipment purchases and personnel training. These costs can be associated with equipment, personnel, facilities and materials; hence, represent a quantitative measure of the costs associated with the system's performance.

Although cost determinations involve a reasonably direct accounting of expenditures, value determinations are a more complex problem. The value of an information system and its costs are not necessarily in proportion nor are they measureable in the same manner. Costs can be quantitatively denoted at every stage of processing from collection to output; however, the value of an information system is connected with user performance and capability which may only be assessed in a qualitative manner.

3. System Response Time

From a system user's perspective, system response time is the period that lapses between the statement of information need and the reception of output satisfying this need. Response time is a function of the number and the nature of the equipment, the efficiency of the man-machine interface, the capability of the operating programs and procedures, the communications capability within the system, and the sensitivity and depth of information representation. System response time is another quantitative measure of a system's performance.

If we can assume that the collection effort satisfies the intelligence requirements, and that the data transformation through the system is nondegrading, then the retrieval effort within the intelligence system can be evaluated with respect to response time and operating costs.

Since the operating time of equipment and personnel are closely associated with operating costs, a time-flow model can be easily modified to provide operating costs of retrieval. Therefore, the first goal in this research effort has been to simulate the response time of mechanized information storage and retrieval systems.

At a later date, refinement and extension of the model could conceivably include the ability to specify cost constraints, response time constraints, basic operating concepts, and state-of-the-art equipment characteristics. Simulation output would be alternative acceptable configurations under the given constraints.

C. BACKGROUND

1. The Initial Model

One way to simulate an information retrieval system is to consider those operations which must be performed by the parts of the system. Certain steps must invariably be followed in obtaining information. These steps constitute time-consuming events. A basic model logic was developed, centered on the response time measure and extendible to any specific or general computer based information retrieval system. This basic logic is illustrated in Figure 1.

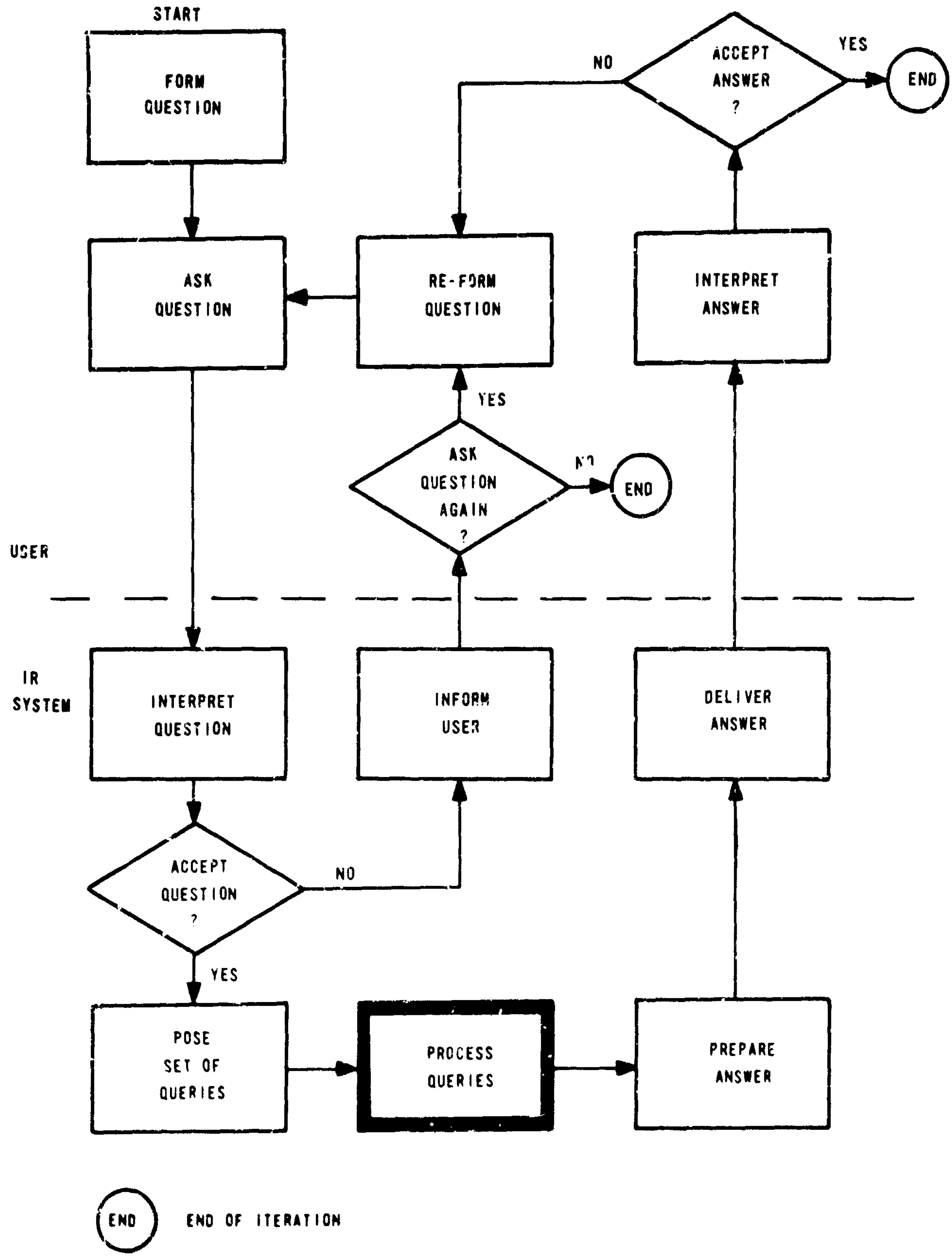


FIG. 1 BASIC MODEL LOGIC

Based upon this logic, a simple simulation model was programmed and operated for research purposes only. This research simulator required two types of input-event times and selection probabilities. The event time data described only the time range for a given event. The selection probability data referred to the observed usage of the various query types and I/O devices. While working with the research model, two factors that needed to be incorporated into the simulation model were immediately quite evident -- (1) an ability to examine equipment characteristics and (2) more freedom in specifying time data. In addition it was also noted that a true response time was not being produced since the program did not consider the effects of processing error, interactions of data flow through the facility or the effects of varying operating schedules, all very strongly interacting elements influencing the response time.

2. The Current Model

Employing the rationale and diagnostics produced by the basic model, the research effort was expanded to create a structure in which an engineer can specify the precise nature and schedule of the specific operations and identify possible points where errors can effect processing.

There are essentially five necessary steps required to define a system within the framework of this research model. In each step, part of the dynamic expected behavior of a system is identified and mapped into the model under the formal language of the simulation program. The parts of a system considered in this effort are as follows:

a. Operations -- What are the time-consuming operations of the systems?

b. Linkages -- What paths do different data follow during information processing?

c. Service Units -- How many service units (devices and/or personnel) are there available at each operation?

d. Availability -- What is the processing schedule of the system? What service units may be down for repair or maintenance?

e. Processing Load -- What is the volume and frequency of the different querytypes that may be placed against the system?

These five aspects of aninformation processing system constitute the basic input categories of the simulation program and these may be manipulated to provide analysis of the system's performance. For example, the saturation point of a given system can be examined by increasing the processing load while holding the remaining variables constant. Once the saturation point has been reached under the given system state, the problems of increasing the system's capacity can be analyzed by holding the increased processing load constant and manipulating the other variables (e.g., number of service units, speed of equipment, etc.)

D. BASIC ASSUMPTIONS AND LIMITATIONS

The present information systems model is intended to reflect the time expended by a mechanized system's response to a user's inquiry. The current model can be described as a topologically structured series of nodes and links which may be assembled in some desired serial fashion to characterize some specific information retrieval system or system concept. Each node represents some time-consuming operation.

Initially it was anticipated that this time-flow simulation concept would only be applicable to computer-based information systems. This assumption was intuitively based on the observation that time-consuming functions of a computer system are consistent (mechanical in nature) and contain observable parameters amenable to measurement. In a previous report,[1] time parameters for such well-defined functions as read time, write time, etc., were formulated and presented for inclusion in the model. It was noted, however, that if time histograms were developed from such time formulas <u>outside the basic program,</u> then the model becomes a generalized simulator capable of depicting the flow through many varied types of mechanized information systems. The use of time histograms, in lieu of time formulas, broadens the application of the model, but increases the requirements for engineering calculations. The utility of the

[1] "An Information Retrieval System Model," October 1965 (AD623590).

simulation is, to a great extent, now dependent upon the engineer's ability to adequately express the distribution of processing time at each event.

Although the present generalized time-flow model has eliminated some initial assumptions about the kinds of systems encompassed, several limitations still exist that restrict the scope of the model; hence may limit the real world domain of systems reflected. These restrictions have been accepted in the present model in order to expedite the testing of the model's feasibility and are as follows:

1. The user essentially interacts with the system at only two points; i.e., he poses a question and receives an answer -- he does not monitor intermediate processing.
2. The user's question initiates only one query.
3. The amount of time consumed by a component performing an assigned task is depicted with time distributions.
4. There are only two kinds of time-consuming events available within the model; i.e.:
    a. One type of event processes all data backlogged in a queue when the event becomes available; thus delay time in queue is a function of event availability and is not a function of the size of the queue.
    b. The second type of event is responsive to only one processing task at a time and and must complete each assigned task before performing the next assignment.[1] The queue unloading strategy essentially is first-come-first-served to the first available service unit.

[1] A messenger picking up the mail at an appointed hour is an example of the first type of event; card verification at one station is an example of the second type of event. At the appointed hour all mail in queue are dispatched for delivery; but a card in line must delay until all preceding cards have been completed.

5. The service unit assignment within an event is from "left to right."

6. The man-machine match is assumed nonrestrictive to query processing. For example, if there are several key punch machines available, it is assumed that there are also sufficient operators available.

7. The processing of a query within the system is a deterministic operation based on probabalistic routing. There is no testing of the state of the environment when assigning datum for processing at an event. For example, if two dissimilar units are available to perform an operation (e.g., a 200- and a 600-line per minute printer), the model does not attempt to optimize the work assignment.

Such assumptions eliminate the ability of a node to perform simultaneous operations or to be interrupted during a task. This limitation may therefore restrict the simulation of time-sharing devices or some human operations.

In addition to these conceptual limitations, there are mechanical limitations in the model that are produced by the size of the memory core of the ADP system selected to process the program. This size limitation imposes boundaries that restrict such things as the number of steps in a query's anticipated processing sequence, the number of nodes that can be depicted and the number of queries that can be processed. There are also restrictive relationships such as the length and number of time increments and the length of the time line. A complete list of these types of limitations can be found in Appendix B at the end of this report.

II. FUNDAMENTAL CONCEPTS

Four basic observations have influenced the development of the general time-flow model:

1. The processing time and the data flow at each operation (event) within the processing system can be influenced by differences among the requests flowing into the system. Some requests, for example, require more extensive file searching than do others; some requests may be dispatched by mail, while others are phoned, etc.

2. Errors in data handling may significantly influence processing time. Error rate may be a function of the operation, the equipment used and/or the type of data being processed.

3. The availability of equipment and personnel are significant factors in the amount of delay accumulated in the response time of a system. For example, the high performance capability of a central processor can be wasted if input is bottled-up at the satellite computer.

4. The interactions of data flowing through the system are significant elements influencing the response times. The degree of influence created by such interactions is a function of the system's load factor and method of assigning the queries to the processing events.

As a result of these observations, four basic concepts for a general time-flow model were postulated. That is, given a specified system or system concept --

1. The general sequence of operations and the time expended at each step may be <u>dependent</u> upon the nature of the request.

2. The deterministic path of a request through a system may be interrupted by errors encountered in processing. The probability of interruption may be <u>independent</u> of the nature of the request.

3. The relative position in time of all system components (e.g., the system user, communications, processing personnel, etc.) must be specified in a meaningful simulation. Moreover, provision should be

made to indicate an estimated unscheduled absence of personnel or the likelihood of equipment failure.

4. The processing work load, event availability and data processing within the system must be integrated under the simulation.

The following discussion illustrates how these fundamental concepts have been embodied within the mechanical framework of the present simulation model.

A. QUERY TYPES AND QUERY GROUPS

Requests posed against a system can be reasonably categorized according to the "paths" they take. Two requests generate different query types if their expected "path" through the system are different; i.e., if their input media, search type, and output media differ. For example, suppose system user A can dispatch his requests by courier or by telephone. The requests sent by courier are processed and returned by courier; however, those sent by phone, depending upon their priority, may be returned by courier or transmitted over a data link. Additionally, assume that user A's requests can be classified as "low" search or "high" search; i.e., the expected file search time ranges between 3-15 minutes (low) and 15-45 minutes (high). Six query categories may be defined for user A as follows:

QUERY CATEGORY	INPUT MEDIA	TYPE OF SEARCH	OUTPUT MEDIA
1	Courier	Low	Courier
2	Courier	High	Courier
3	Telephone	Low	Courier
4	Telephone	High	Courier
5	Telephone	Low	Data Link
6	Telephone	High	Data Link

Sometimes query types are created by a simple desire to differentiate among many queries with identical processing paths, but initiated by different user groups. For this reason and various other assorted criteria, it is quite possible that the output data generated by a particular series of queries would be desired summarized as a group. Such associations among query types and interests in collective data give rise to the generic classification of query groups. A query group, then, is a collection of specified query types that have some common basis for association, and is employed to simplify the output of very large numbers of query types, or simplify initial user group identifications.

Regardless of the categorization made, all paths taken have common attributes; i.e., events must occur, time consumed, and success/failure established. More formally, the query path is operated upon by:

1. Event Sequence -- the order and nature of the operations required to receive, process and deliver the completed request to the user.
2. Processing time -- the expected processing time required for a particular request at each event.
3. Processing failure -- the likelihood that a processing step will be unsuccessful and the query will be interrupted from its normal flow.

Each of these operations are discussed more fully below.

1. Event Sequence

Figure 2 presents a simple flow diagram of part of a computer-based processing system. The operations or events have been numbered 10-80; the solid lines indicate normal flow, the dashed lines indicate error routes. Imbedded within this flow are several explicit and implicit aspects of the general model. First -- the quantity and nature of the events are specified by the investigating engineer. Second -- the placement and the level of acceptance of error tests are specified by the investigating engineer. Third -- the level of acceptance of error tests may be the same for all queries or may be dependent upon the query type. For example, in the system depicted in Figure 2, 6% of all punched cards will be rejected; however, the decision to correct

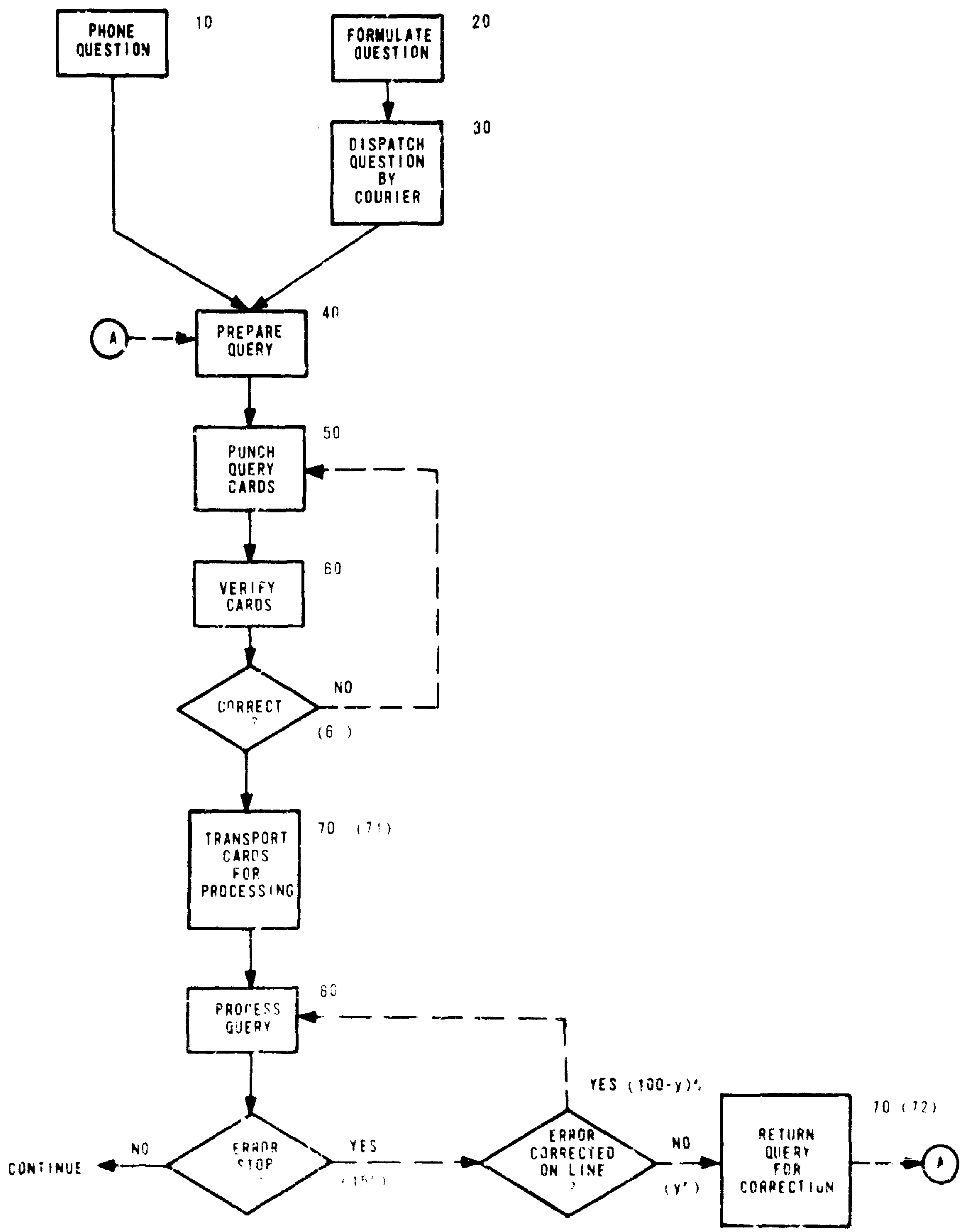


FIG 2 EVENT SEQUENCE

errors on-line will depend upon the nature of the request.[1] A "high priority" request may have an on-line correction threshold of 90%; whereas a "low priority" request may have only 40% of the error steps corrected on-line.[2] A fourth point illustrated in the simple flow diagram is that different operations may have the same basic event number. Event 71 (transport cards for processing) and event 72 (return query for correction) are interconnected in the sense that one individual or group performs both operations. If 71 is transporting cards, 72 cannot simultaneously return a query for correction unless there are two or more operators and at least one is available. Fifth -- an operation may, in fact, represent a complex of operations. For example, event 80 (PROCESS QUERY) can be defined to consist of the following operations.

a. Operator setup and query entry

b. File search

c. Plus ONE of the following substrings,

   (1) Record sort

   (2) Record sort and edit

   (3) Record sort, edit and summary.

The selection of a substring in event 80 can be dependent upon the type of request being processed through event 80. On the other hand, it is possible for the investigating engineer to specify a selection probability independent of the query type; e.g., let "a" be selected 10% of the time, "b" selected 70% and let "c" be selected 20% of the time for all requests.

[1] The occurrence of errors is treated as a random function in the simulation program. At each test point, a random number is generated and tested against the specified threshold value.

[2] It should also be noted that the average number of queries returned for correction is expressed as (.15) (y) N where N is the total number of queries flowing through event 80. Thus, if N = 100 and y = 60%, nine queries (on the average) will be returned.

When error test points are interjected into the flow diagram, the path of query type becomes probabalistic; e.g., the normal event string for query type 1 (from user A) through the system illustrated in Figure 2 is:

20, 30, 40, 50, 60, 70 (71), 80, CONTINUE

Other possible strings are:

20, 30, 40, 50, 60, 50, 60, 70 (71), 80 CONTINUE

20, 30, 40, 50, 60, 70 (71), 80, 80, CONTINUE

20, 30, 40, 50, 60, 70 (71), 80, 70 (72), 40, 50, ..., 80, CONTINUE

In theory, (both in the real-world and in the general time-flow model) infinite strings are possible; in practice, however, they are unlikely. The probability that a request will oscillate between events 50 and 60 N times is $(.06)^N$; the probability of a request oscillating three times in this loop is .000216 (about one-fourth the likelihood of drawing a full-house in a poker game).

2. Processing Time

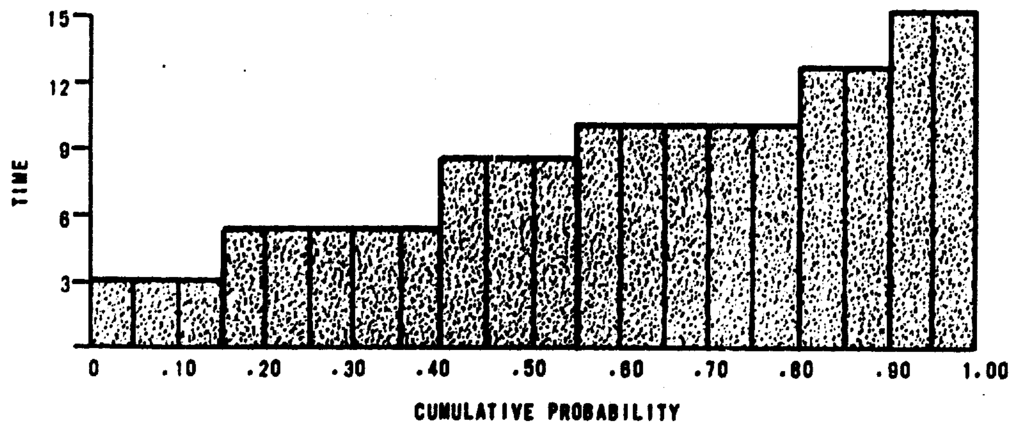
The processing time at each step in a query's path is determined by selecting a value from a time distribution table that is associated with the query type and the operating event. In the above example, a low file search for user A (at event 80 -- PROCESS QUERY) could have a value selected from a range of, say 3-15 minutes; while a high search could choose a value from another range, say 15-45 minutes. It is possible to have several time distributions for the same event associated with different query types, e.g.,

USER	SEARCH	RANGE IN TIME
B	LOW	1-10
B	HIGH	10-20
C	LOW	1- 5
C	MODERATE	5-15
C	HIGH	15-30

Moreover, it is possible to have different ranges of time at the same "event" for the same query type; e.g., one time range for the initial operations and another for error repetitions.

The distribution of a time range is approximated by a discrete cumulative probability distribution partitioned into twenty equal segments (i.e., an ogive having twenty equal parts). For example, the time values within the "low" search interval of 3-15 minutes could be represented by the following associated probabilities:

TIME	PROBABILITY	CUMULATIVE PROBABILITY
3 MINUTES	.15	.15
5	.25	.40
8	.15	.55
10	.25	.80
13	.10	.90
15	.10	1.00



The time selection process in the program is fairly straightforward. As previously mentioned, it was felt that the time values at every 5% probability interval would be sufficiently accurate for obtaining processing times. Therefore, an ogive can be viewed as being composed of 20 individual cells, with each cell containing some given processing time. By generating a random integer number R ($1 \leq R \leq 20$), the address of a cell in the ogive housing the proper processing time is selected.

Frequently the expected processing time at one event is proportional to the expected time expenditure in some other operation. A 600-line per minute printer, for example, is three times as fast as a 200-line per minute device; a 100-word per minute teletype operates at approximately 1.7 times the speed of a 60-wpm device, etc. The time value selected from an ogive can be multiplied by some constant to produce the time expenditure for another operation.[1]

3. Processing Failure

The normal processing path of a query through a system can be disrupted by "unscheduled" occurrences within the system. These occurrences include the effects of errors encountered at some processing step and the problems associated with component failure and maintenance.

The error rate for an operation may be dependent upon the nature of the data being processed (e.g., tape redundancy stops are somewhat proportional to the volume of high speed tape passage.) On the other hand, error rate may be a function of the operation, independent of the data (e.g., "noise" picked-up in teletype transmission is somewhat a function of the atmospheric conditions and not dependent upon the data being transmitted). In the present simulation, a threshold value can be specified as an error probability to accept or reject processing at an event. One error probability can be specified for all processing at an event; or different probabilities can be specified according to the different query types. Thus, probable processing malfunctions can be a function of the processing step or a function of the nature of the data being processed.

Presently, component failure is specified as the probability that a service unit (or operator) will fail in any given time interval. A component that fails is down for a specified fixed time interval. This is somewhat unrealistic;

[1] Since all time expressions within the simulation must be in the same units (e.g., seconds, minutes, hours, etc.), it is sometimes necessary to select the most frequently encountered unit as a base and multiply the developed ogives to obtain larger (or smaller) time values.

however, it provides a simple method of examining some of the effects of component failure.[1]

B. LOAD FACTOR

The number of questions arriving at a facility for processing within a given period of time, coupled with the amount of time required to process the generated query to some acceptable end point, represents the operating burden on that system. This burden is defined as the system's load factor. In "real world" situations this is a dynamic factor since there are a variety of stochastic processes that when collapsed together determine the load. Therefore, normally no predeterminable figure representing exact arrival numbers and associated processing times is calculable. Within this model the simulated load factor is not a deterministic quantity either, but rather a function of values determined from different probability distributions, different integrable events that can accrue various delay times, and varying facility or elements of the facility and user availability times.

Even though the system's load factor is dependent upon many stochastic factors, an engineer can approximate these factors by defining:

1. The number of questions posed against the system.
2. The type of query initiated by each question asked.
3. The initiation frequency of each type of query.

These elements, taken together, define the load placed against the simulated processing effort.

C. PROCESSING SCHEDULE

The basic criterion for evaluation of an IR system in this research effort is time. Therefore, a continuous straight line (called the time line) is used

---

[1] Another approach would be to let the probability of failure be a function of the amount of component usage since the last repair or scheduled servicing. The down time could be a function of the time of failure (reflecting the availability of maintenance personnel) and the probable requirements for repair.

here to depict a segment in the operating life of an information system. The length of the segment can be some appropriate unit of time such as a day, a week, a month, etc. Associated with the time line are two important relative time segments; i.e.:

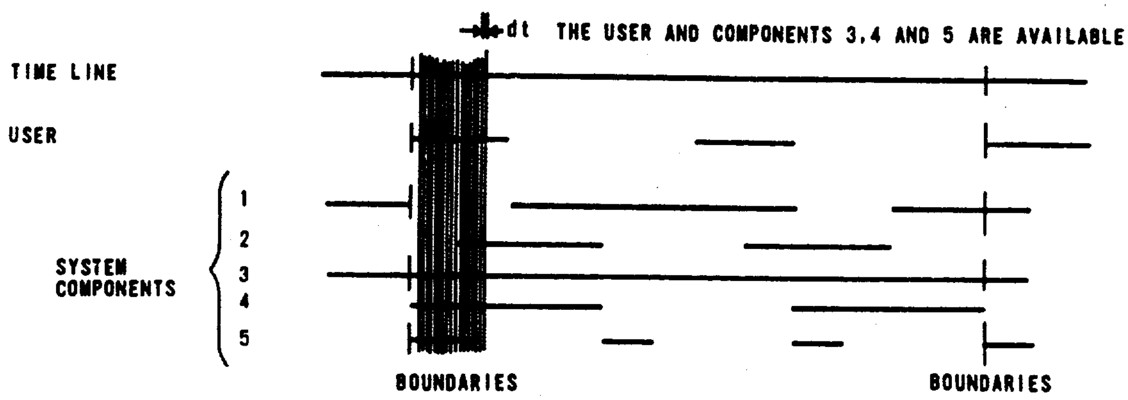
1. User's availability time -- intervals that a system user (i.e., one seeking information from the system) is available to pose inquiries or receive output data.

2. Facility's availability time -- intervals that a processing facility (or any component within the facility) is available to process information requests.

These availability times need not be continuous. Therefore, the user and facility availability time segments can be treated as being subintervals positioned relative to the time line. These subintervals may coincide, overlap or they may be disjoint.

Imbedded within this line are segments that represent relative positions corresponding to states in the processing environment. These states of facility processing include such aspects as the availability of individual equipment or service elements within the facility, the user's work schedule, as well as load factors such as the arrival of queries. Any meaningful real world approximation of an information system requires a method of relating the unique interactions caused by state changes over the interval of time being simulated. It is, therefore, necessary to identify both the relative position of the states within the simulation as well as the location of "current time" along the time line during the simulation.

One method of identifying the relative state of individual components is to define points on the time line that represent the range of the influence of the particular state. For example, we could specify that between 0830 and 0945 event 20 is not available for processing. Then while incrementing the "current time" over the time line by very small homogeneous dt segments, the state of all components can be examined at every consecutive dt segment. This concept is illustrated in the following diagram:



The summation of the discrete data generated within each differential quantity over the selected time line provides the simulation output. This approach, while theoretically sound, was considered to impose an excessive processing requirement on the research effort. For example, the simulation of only one day day in dt segments of 1 minute would require 1,440 incrementations. This approach becomes even more complex as the chosen simulation segment unit of time is increased or the dt segment length is reduced.

A potentially more efficient variation of the previously outlined approach would be to increment the simulation by the greatest common divisor of homogeneous grouping of dt's. Thus, if 12 one-minute dt's constitute the greatest common divisor of homogeneous time segments, then 120 incrementations would be required to simulate one day. This approach, however, could produce time increments of varying length which might conceivably approach the length of a dt segment from time to time. Moreover, if component availability can vary between iterations in the simulation (e.g., a component made unavailable under a failure probability), then the greatest common divisor would have to be recalculated each iteration. Thus, the variation is not necessarily always an improvement over the first approach.

Through a series of trade-offs, a mechanical technique of subdividing the time line into homogeneous states has been adopted which represents a compromise between providing increased resolution and overburdening machine processing. The subdivisions are referred to as Δt's and have the following properties and restrictions:

1. All Δt's span the same amount of time.

2. A Δt must be a divisor of the time line; i.e., the total simulation time divided by the amount of time per Δt must yield a whole number.

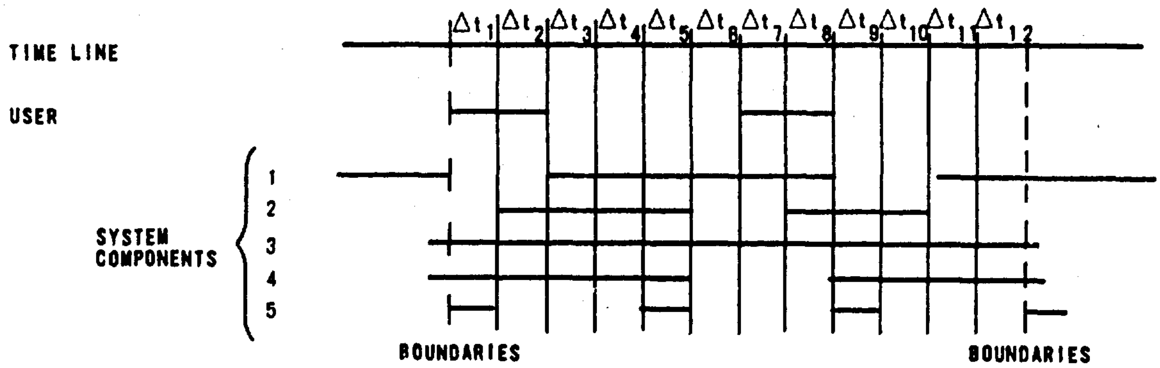
3. The length of a Δt can be specified by the investigating engineer; however, a time line can be partitioned into at most 400 Δt segments.

4. A Δt must be defined in terms of the basic unit of time for the simulation.

5. The state of the users and the system components must be consistent for an integral number of Δt's.

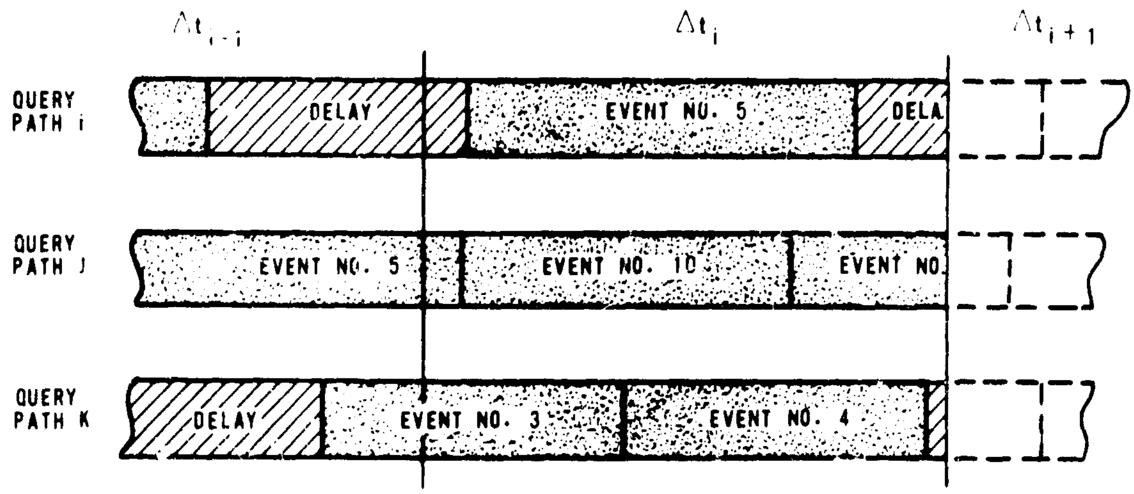
Numbering the Δt intervals consecutively provides a very simple method of relating important variations for the simulator. For instance, an engineer can identify the components scheduled over a range of Δt's; he can designate different processing loading factors over different grouping of Δt. The smaller a Δt is defined, the closer it approximates the concept of a dt increment. Thus, the degree of compromise between the resolution of the model and the processing burden is at the discretion of the systems engineer.

This concept of the time line and Δt's is illustrated in the following diagram:



NOTE: ALTHOUGH THE USER AND FACILITY TIME LINES NEED NOT BE CONTINUOUS, THEY ARE EITHER TOTALLY AVAILABLE OR TOTALLY UNAVAILABLE WITHIN A Δt INTERVAL.

The simulation program extends the processing path of each query in the IR system during a Δt interval by the processing times (or appropriate delay times) accrued by each query until all paths are updated to a Δt boundary. Then the states for the next Δt interval are determined and the procedure is repeated. This process is illustrated in the following diagram:



Thus a busy component that becomes unavailable during the next Δt will delay the query being processed (and all queries in queue) by a factor of Δt.

III. AN EXAMPLE SIMULATION STUDY

The material in this chapter presents an example simulation study of a computer based information system. The presentation is made in four parts; each part corresponding to a definite phase in the engineering effort, i.e.:

1. problem definition,
2. system definition,
3. parameter expression, and
4. output examination.

This example, while essentially realistic, does not reflect an existing system; but is, instead, a reflection of methods and components used in a family of contemporary systems. To a great extent, the "system" portrayed in this study represents the kinds of computer systems now being considered in support of Naval Intelligence analysis efforts.

A. PROBLEM DEFINITION

This first phase in the simulation effort is not a requirement of the simulation, but is recommended as an aid in establishing the analysis and evaluation criteria that will be used in subsequent efforts. In this first step, the scope of the system problem should be identified. Are we building a system, expanding one, modifying some of the components, testing a system against new or different requirements, etc.? At this point, the performance criteria for the system should also be defined. What operating characteristics are essential; which are desirable?

In this example study, a system configuration exists and the problem is to utilize these components to satisfy a new processing requirement. Figure 3 illustrates the components of the "existing" system. Currently, the processing requirements for the system are to:

1. Process (on a daily basis) a group of intelligence reports received from various sources.

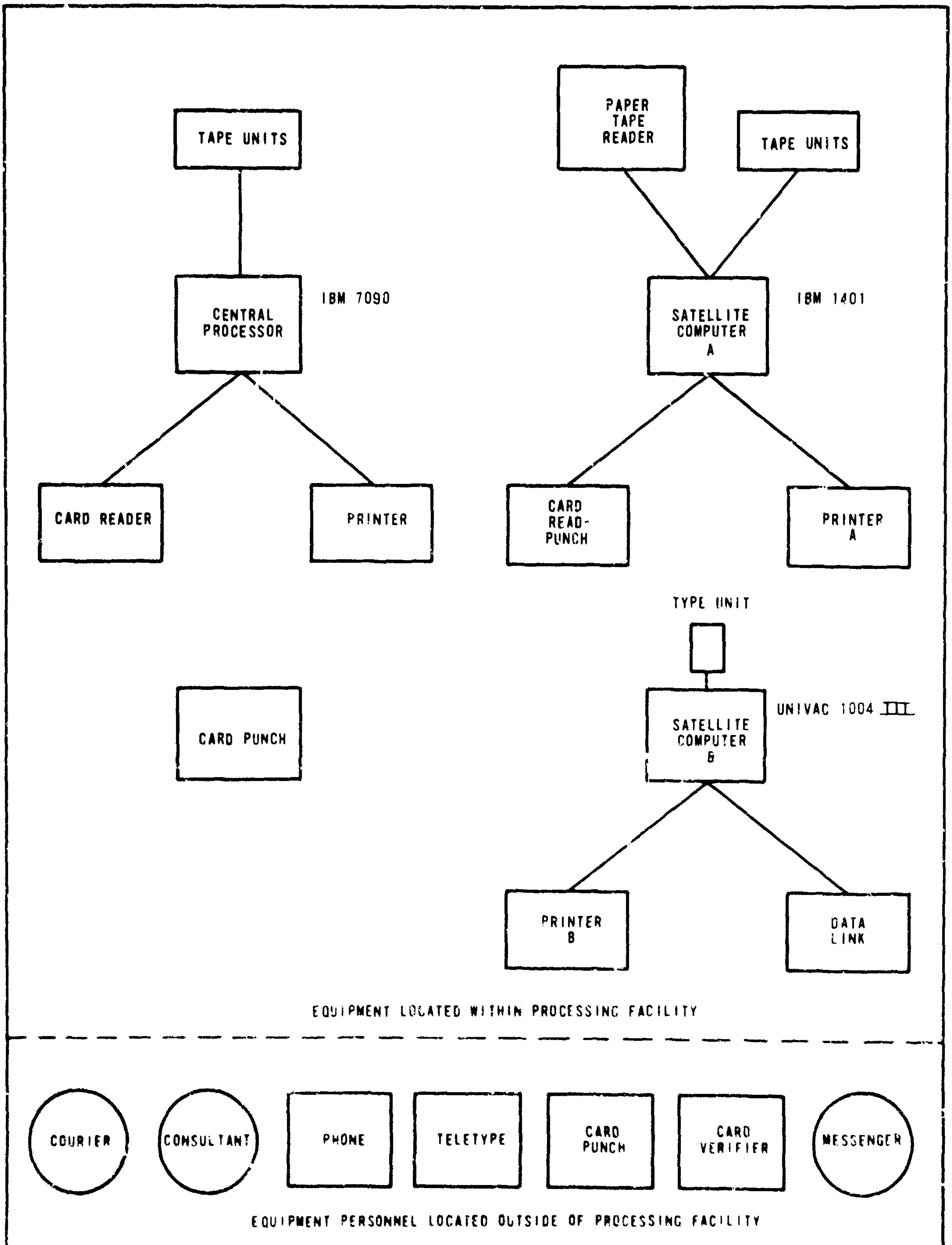


FIG 3 COMPONENTS OF EXAMPLE SYSTEM

2. Summarize the contents of these reports and provide weekly and monthly activity output listings.

The new requirement levied on the system is:

3. Provide output in response to spot inquiries posed by the system users.

Under the new requirement, it is required that both the system response time and processing costs be as small as possible.

In this example, the system components and processing priorities are fixed.[1] The design engineer may, however, (1) exploit different I/O techniques using existing lines of communications between the user and the processing facility and (2) vary the processing schedule of the system components. Changes in the system must not, however, reduce the quality or timeliness of the present report production capability.

B. SYSTEM DEFINITION

The second phase in the simulation effort requires that the engineer (1) identify the time consuming functions (events) of the processing effort and (2) describe the expected flow of data through these events. This phase can probably be best accomplished with the aid of a system's flow diagram, processing schedule and a work load schedule.

1. Flow Diagram

Figure 4 illustrates a flow diagram of the processing events in one proposed input/output design. There are three input routes that an inquiry can take into the system: i.e.:

a. A user can phone a question and a system consultant will prepare a formal request (query). This method is recommended for all complex requests.

---

[1]Processing priorities, under the new requirements, are as follows:

1. report production,
2. file updating,
3. input processing,
4. information retrieval,
5. new programming efforts.

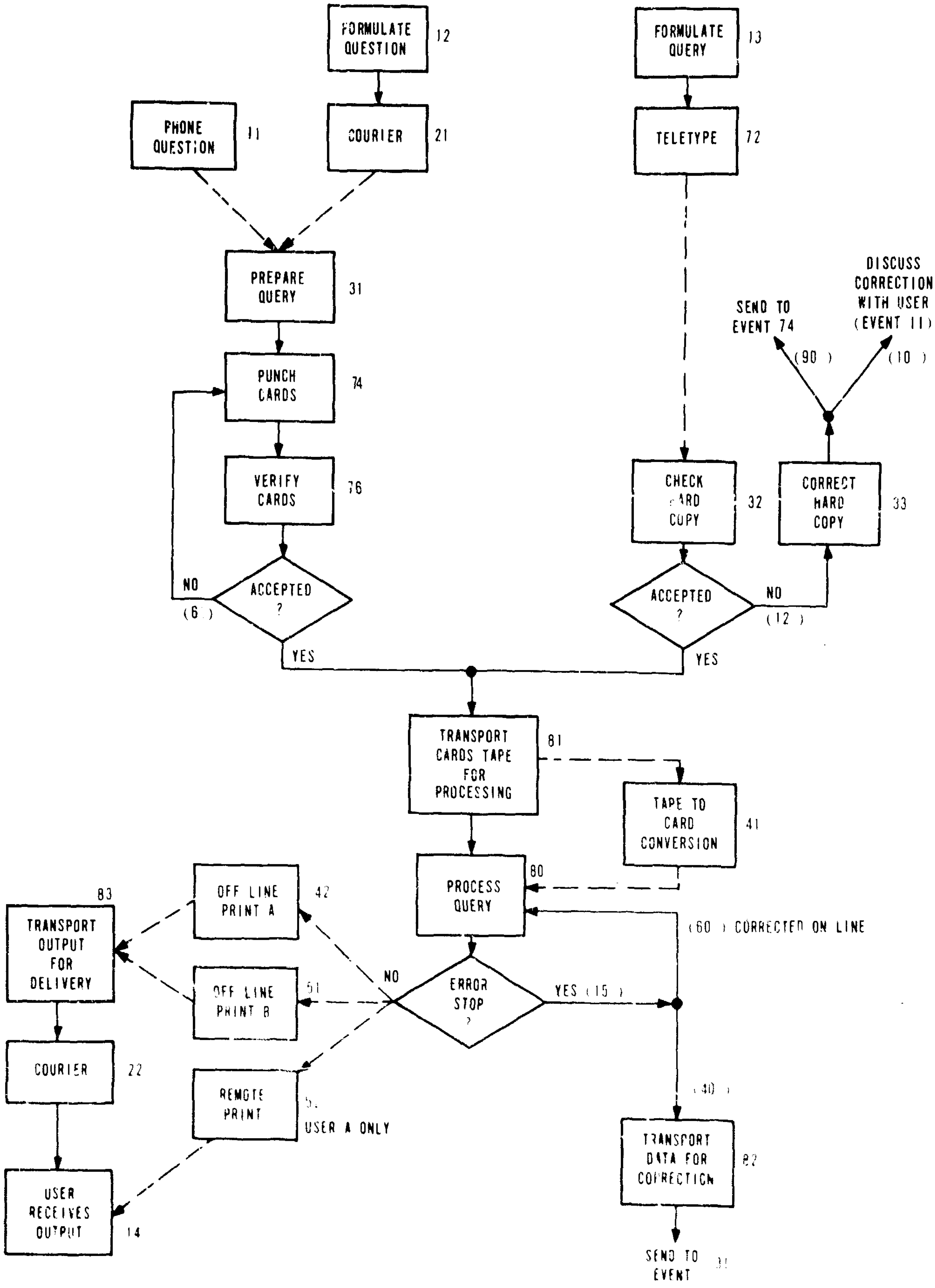


FIG. 4 PROCESSING EVENTS

b. A user can dispatch a question by courier service. This method is recommended for simple low priority inquiries.

c. A user can dispatch a query be teletype. This method is recommended for simple high priority inquiries.

There are two output routes that can be taken to transmit data to the users; i. e.:

a. Courier service; recommended for all priority inquiries.

b. Remote print via data link; recommended for all high priority requests.

Unique points in the proposed design where error may affect the processing effort have also been identified. For example, it has been estimated that 15% of all the queries will encounter some difficulty in computer processing.[1] Of this problem set, it is predicted that 60% of the difficulty will be simple and can be corrected on-line. The remaining queries in the problem set (.15 x .40 = 6% of all queries), however, will be returned for correction.

Each processing event has been given a numerical label for simulation identification. Part of the input into the simulation program (see Appendix B, Example Simulation I/O Displays) lists these labels with a short description of the event as well as the number of service units available, the probability that a service unit will fail in any given Δt, and the event number of any other event interlocked with this event. For example, in the following listing, two teletypes are available for receiving queries and there is a 2% probability that one unit will fail in any given Δt interval. Additionally, there are two messengers who perform three functions; one being to transport cards or tape into the processing facility, another being to transport data for correction and the third is to transport output for delivery.

---

[1] Tape redundancy halt, jammed card in the card reader, etc., are some examples of the difficulty that can be anticipated in processing. The percentile used in reflecting these difficulties essentially reflects the previous history of the facility.

Event Code	Meaning	Units	Main. Prob.	Locked to
. . .				
72	Teletype	2	2	0
. . .				
80	Messenger	2	1	0
81	Transport cards/ tape for processing	2	0	80
82	Transport data for correction	2	0	80
83	Transport output for delivery	2	0	80

2. Processing Schedule

A processing schedule is simply a representation of the planned availability of each system component and the system users. In the present version of the simulation, the schedule is depicted by intervals of Δt over the simulation time line; e.g.:

FROM	TO	EVENTS AVAILABLE
1	4	72 50
5	6	72 50 40
7	7	72 50 40 60

During $\Delta t1$, 2, 3 and 4, events 72 and 50 have been scheduled (the teletype and satellite computer B) for processing support. During time period $\Delta t = 7$, events 72, 50, 40 and 60 are scheduled to be available.

Figure 5 depicts a possible facility schedule for information retrieval processing under the proposed input/output design.[1] For the example study, this schedule means that information retrieval processing will have top priority at each event during the time indicated; moreover, IR processing will only be accomplished during these time intervals.

[1] In many circumstances one would not schedule specific components to be available for specific jobs; but would, instead, schedule the overall system and assign work priorities on the jobs (see Chapter IV RECOMMENDATIONS).

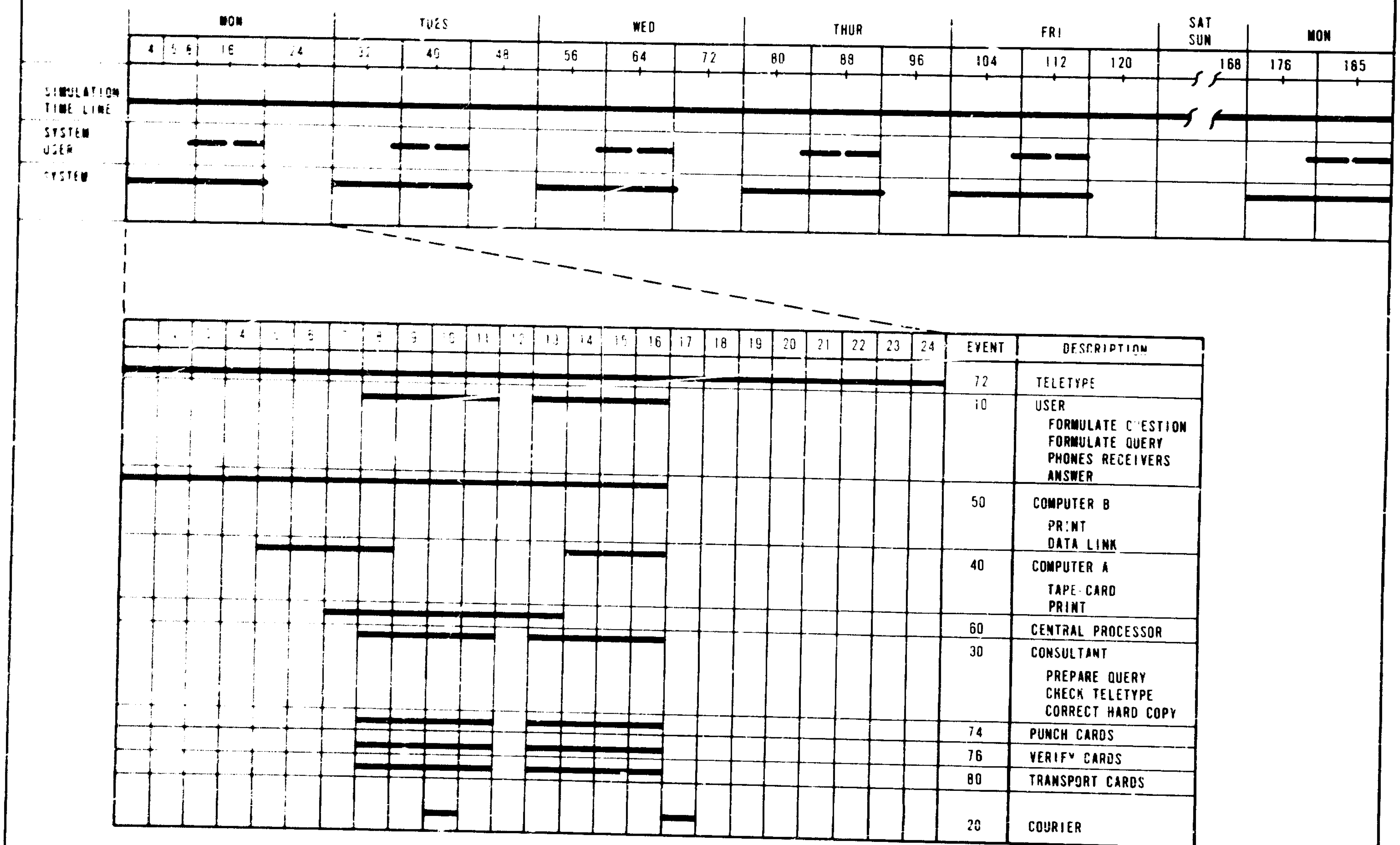


FIG 5 PROCESSING SCHEDULE ONE

The period selected for simulation is, essentially, one working week. During this period, the system users work an eight-hour day, five days a week. The processing system is available two eight-hour shifts, five days a week. In this example, the component schedule for IR processing is the same for each work day.

At this point one might ask, is the input/output design and the processing schedule a "good" one for the given system configuration and processing requirements? Both the design and the schedule are products of an engineer's concept of a workable solution. Neither, however, have been subjected to any sort of objective testing. One method of testing this concept is to simulate the IR processing of the expected inquiry load over the simulation time line, examining the flow for bottlenecks, and assessing the response time and processing costs of the IR effort.[1]

3. Work Load Schedule

Under the present simulation, work load schedule is equivalent to the query loading factor. Ideally, however, the work load schedule would reflect the priority and the expected influx of different distinctive job types into the processing system, e.g.,

a. report production,

b. file updating,

c. input processing,

d. query processing, etc.

The query loading factor reflects the different types of queries and the distribution of these different types over a specified interval of time. This distribution in time can represent the arrival of the queries at the processing facility, the posing of the requests by the user, or any meaningful initiation of the processing effort.

---

[1] In this study, processing costs will be measured in terms of component utilization.

a. Query Type

This is a pragmatic distinction made among queries in the sense that two queries are of different types if we can expect that they will follow different routes through the system or will consume different amounts of time at the same event. The following table represents the query types defined in the example study.

QUERY CODE	MEANING			
	SITE	METHOD OF INQUIRY	SEARCH	OUTPUT COMMUNICATIONS
1101	A	Telephone	Low	Courier/Data Link
1102	A	Courier	Low	Courier
1201	A	Telephone	High	Courier/Data Link
1202	A	Courier	High	Courier
2201	B	Telephone	High	Courier/Data Link
2202	B	Courier	High	Courier
2303	B	Teletype	Lo-Hi	Data Link
3101	C	Telephone	Low	Courier
3102	C	Courier	Low	Courier
3303	C	Teletype	Lo-Hi	Data Link

In the above table, SITE represents the locations of different members of the user population. This distinction is particularly significant in considering the time required for the courier to deliver requests and system outputs.

METHOD OF INQUIRY distinguishes the input routes of the requests.

SEARCH categorizes the expected computer processing into three levels of difficulty. Specific distributions for the different computer operations (e.g., file search, sort, edit, print, etc.) will reflect the degree of processing difficulty in these three levels.

OUTPUT COMMUNICATIONS distinguishes the expected output routes of the requests. Since all teletype requests are "high priority," all teletyped requests will have their responses dispatched by data link. Similarly, all courier requests are low priority; therefore, their responses will be dispatched by courier. Telephone requests are complex and may be either high or low priority; hence, their responses are dispatched over one or the other media. The selection of an output route is a probabilistic decision in the simulation. The probability that an output will be dispatched by courier is equal to the probability that the request is of low priority.

There are two points that should be made at this time. First -- the meaning of the query type is defined by the investigating engineer; it is not constant from study to study. Second -- the query code (e.g., 1101) is also assigned by the investigating engineer. An important aspect of the query code assignment is that the first two digits (e.g., 11) identify the query group in the analysis program. Query response time (work time plus delay time) is depicted by query groups -- not by query types. In this example study, query groups correspond to SITE and SEARCH; i.e.:

GROUP	SITE	SEARCH
11	A	Low
12	A	High
22	B	High
23	B	Lo-Hi
31	C	Low
33	C	Lo-Hi

Another (and perhaps more useful) query grouping could have been to have let each group contain only one type. Thus, each difference in expected processing would have been explicitly reflected in the output analysis summaries.

b. Query Type Distribution

The distribution of query types over time can be expressed in a uniform or normal distribution or combinations of both. Appendix B illustrates the EXPECTED ARRIVAL OF QUERIES for the example under study. The following Table is a brief excerpt from this input.

FROM	TO		Q	N	N	Q	N	N	Q	N	N
8	11	U	1101	0	1	1201	0	1	1102	0	1
8	11	U	1202	0	1	2201	0	1	2202	0	1
8	11	U	2303	0	2	3101	0	1	3102	0	1
8	11	U	3303	0	2						

The entries indicate that the queries are to be selected from a uniform distribution over the time intervals covered by Δt's 8 to 11. In this interval, 0 or 1 type 1101 query is to be selected; 0, 1 or 2 type 3303 queries are to be selected, etc.

In general, the range in values and the number of time intervals considered determine the number of queries of each type that are generated over the user's time frame. In the above table, for example, the probabilities for generating a group 11 query between Δt's 8 and 11 are:

NR QUERIES	PROBABILITY
0	.25
1	.50
2	.25

If the same range (0-1) had been specified between Δt's 8-9 and again in 10-11, then the possible number of queries generated would have doubled and the selection probabilities for group 11 would have been:

NR QUERIES	PROBABILITY
0	.0625
1	.2500
2	.3750
3	.2500
4	.0625

On the other hand, if the range had been doubled (0-2) over the original Δt interval (8-11), the selection probabilities for group 11 would have been:

NR QUERIES	PROBABILITY
0	.1111
1	.2222
2	.3333
3	.2222
4	.1111

Once a query type is selected from an interval, the arrival time at the first event in the query's processing path is selected at random over the interval of time considered. In this example study, the first event for all query types is the user initiating the request; thus the query arrival time is not at the facility, but is the start of the request with the system user.

C. PARAMETER EXPRESSION

This phase of the engineering task is perhaps both the most critical and the most difficult part of the simulation effort. In this phase, the expected path of each query is defined. There are, in general, two distinctive aspects of this definition, i.e.:

1. Identification of the necessary processing events for the different query types. This may include both deterministic and probablistic processing flow, as well as the identification of error points and alternative processing routes for each query type.

2. Specification of the time distributions for each query type at each event.

The following discussion briefly illustrates these aspects of the simulation effort.

1. Processing Flow

Figure 4 (Section B of this chapter) illustrates the anticipated flow in the proposed retrieval processing. The basic flow starts with the user, i.e.,

EVENT NR	EVENT
11	Phone Question
12	Formulate Question
13	Formulate Query

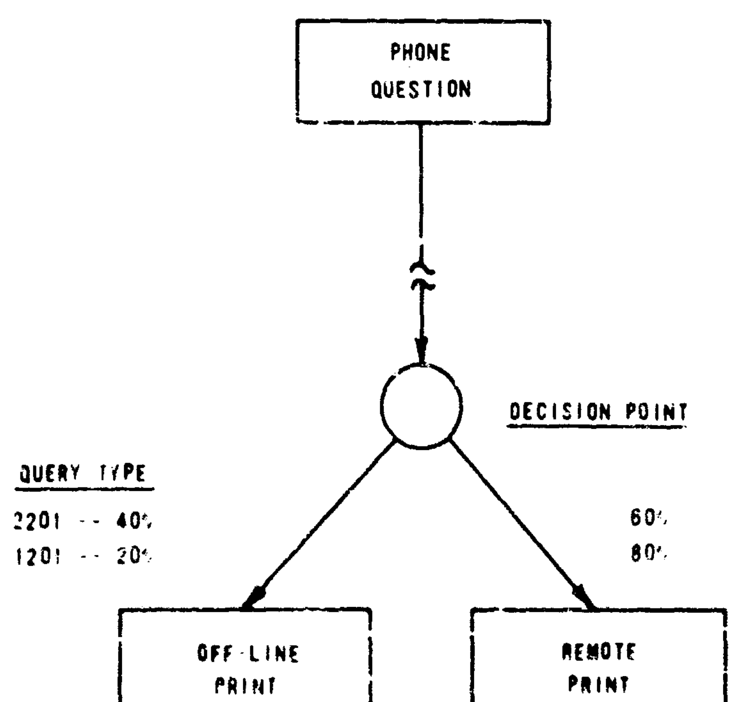
and ends with the system user (EVENT 14, User Receives Output). Under this processing flow expression, response time will reflect all work time and delay time between the start of a question and the receipt of the output. The basic events depicted in this flow chart identify the discrete time-consuming functions that are required in processing the different requests. Alternate processing paths, in this example, are a function of (1) the priority and the logical complexity of the requests, (2) the complexity of the computer processing effort and (3) the errors encountered in the overall processing effort.

a. Function of the Requests

The basic input/output processing path is a function of the query priority (HIGH or LOW) and the logical expression (SIMPLE or COMPLEX); i.e.:

	PRIORITY			
	HIGH	LOW		
INPUT	PHONE	PHONE	COMPLEX	EXPRESSION
OUTPUT	DATA LINK	COURIER		
INPUT	TELETYPE	COURIER	SIMPLE	
OUTPUT	DATA LINK	COURIER		

In this proposed design, all complex requests are discussed with a consultant over a telephone linkage; all requests having a HIGH priority have their output dispatched by DATA LINK. Therefore, in simulating this design, all COMPLEX requests start with EVENT 11 -- USER REQUESTS DATA BY PHONE. The number and frequency of these requests from each user is determined by an analysis of the user's requirements for data support. The probability that a COMPLEX request will be returned by DATA LINK is denoted by the conditional probability (for each user) that a COMPLEX request will be of HIGH priority. Thus, we may have a decision point in the processing flow where the output medium is selected for a query type, e.g.:

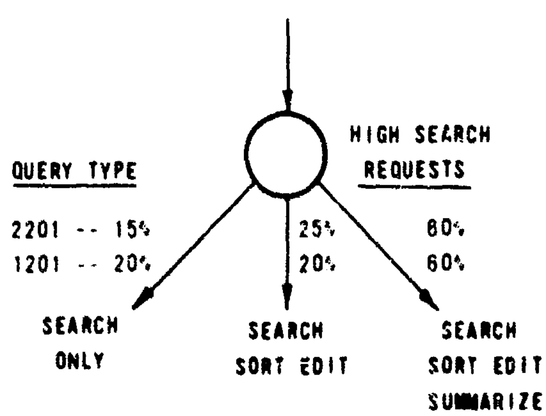


b. Function of Computer Processing

The time expended by the central processing unit has been assumed to be dependent upon the extent of the file search and the degree of processing required. Additionally, the time at the CPU has been assumed to be independent of the input/output routing. In this study, the degree of the retrieval effort at the CPU is categorized as consisting of one of six processing strings, i.e.:

STRING	LOW SEARCH	HIGH SEARCH	SORT AND EDIT	SUMMARIZE
1	X			
2	X		X	
3	X		X	X
4		X		
5		X	X	
6		X	X	X

Query type, in this study, indicates the extent of the file search (LOW or HIGH) but does not denote the degree of computer processing required. The degree of processing has been defined as a variable that is independent of both query priority and the extent of search required. The probability of "string selection" reflects the user's requirements for output presentation. Within the processing flow, a decision point is used to select one of the three substrings available to each query type; e.g.:



The analyses of the kinds of data flow expected in the retrieval effort has linked the system user, the complexity of the search effort, and the complexity of the requests into a definition of query groups and query types. In some instances, the priority of a request is specified under query type; in other instances, however, the priority is imbedded within a decision point. In all cases, the degree of processing at the CPU is a probabilistic consideration. The following table illustrates these interrelationships and summarizes the expected queries and processing flow associated with system user A.

LEVEL OF FILE SEARCH	LOW (Query Group 11)		HIGH (Query Group 12)	
LOGICAL DIFFICULTY	COMPLEX (Type 1101)	SIMPLE (Type 1102)	COMPLEX (Type 1201)	SIMPLE (Type 1202)
REQUEST RANGE Mon.	0-1	0-1	0-1	0-1
Tues.	0-2	0	0-2	0
Wed.	0-2	0-2	0-2	0-1
Thurs.	0-4	0-2	0-2	0-2
Fri.	0-2	0	0	0
PRIORITY	50% HIGH 50% LOW	0% HIGH 100% LOW	80% HIGH 20% LOW	0% HIGH 100% LOW
USE OF CPU				
Search Only	10%	10%	20%	10%
Search, Sort and Edit	30%	20%	20%	10%
Full Processing	60%	70%	60%	80%

c. Function of Error Correction and Detection

Errors encountered in processing may interrupt the normal flow of data through a system. The extent of the interruption will usually be a function of both the severity of the error and the point of detection.

In this study, three error detection points have been identified. Associated with each of these is a probability that the processing effort will halt at that point for corrective action.[1] These error check points are:

PUNCH CARD VERIFICATION	6% failure
TELETYPE VISUAL SCAN	12% failure
COMPUTER STOP	15% failure

The failure rates essentially reflect design experience with the equipment-personnel-operation identified within the system.

The nature and difficulty of the corrective action associated with these failures is depicted by both the routing of the data at the error check point and the time used to correct the error. In this study, for example, 10% of the errors encountered in the TELETYPE VISUAL SCAN have to be discussed with the user; 90% will be corrected by repunching the query statement. Error time expenditures will be discussed in the next section.

2. Time Distributions

The time expended at each step in the processing effort is, essentially, a function of (1) the component (operator-device) used in an operation and (2) the complexity and volume of the data being processed. Within the time-flow simulation concept, these two functions are brought together by defining relationships among the processing event, the query type and the time distributions. The

[1] These failure rates have been specified to be independent of the query types, though it is possible to specify different failure rates for different query types.

following discussion illustrates this relationship by examining a specific problem under the example study, i.e., the estimation of the file search time expended in computer processing.[1]

FILE SEARCH TIME

The system depicted in this study has the following processing characteristics.

CPU	IMB-7090.
TAPE UNITS	IBM 729 VI; read/write 112.5 inches/sec.
MAGNETIC TAPE	2400 foot reels; 800 characters/inch.
LOGICAL RECORD SIZE	Each record contains 360 fixed characters plus a variable field estimated to average 128 characters.
BLOCKING FACTOR	Maximum block size is 12,000 characters.
READ/WRITE	Operations are overlapped with essentially a nonstop read capability.
BATCHING	Requests are not batched.

The general file search time expression is

$$T_{FS} = ST + (T_R + T_W + T_{RC} + T_{RW}) + T_I$$

where

- ST is operator set-up time.
- T_R is the time required to read the file.
- T_W is nonoverlapped write time.
- T_{RC} is recovery time; i.e., time required by the program to read or write past a tape redundancy stop.
- T_{RW} is tape rewind time for the output tape feeding the next processing phase (e.g., sorting).
- T_I is the internal processing time required in excess of the read time.

---

[1]A more extensive discussion of processing time formulae, pertinent to computer based IR systems, can be found in HRB-Singer Report 352.14-R-1, "An Information Retrieval Model," 1965 (AD 623 590).

In this instance, both $T_W = 0$ and $T_I = 0$; moreover, T_{RC} may be depicted under the "on-line" error correction operation; thus

$$T_{FS} = ST + T_R + T_{RW} .$$

Operate set-up time in this facility is fairly constant. The only chargeable time in the operation is the time required to find, mount and dial the first two tapes of the search. Up to eight tapes can be mounted in the input channel at one time; thus, any remaining tapes required in the file search can be mounted while the first tapes are being processed. Observation indicates that ST = 2 minutes (approximately).

The time required to read the file is a function of the length of tape read; which, in turn, is a function of the number of records in the file. A full reel (2400 ft.) of tape requires 4.27 minutes processing time in a nonstop read operation, i.e.,

$$\frac{(2400 \text{ ft/reel})(12 \text{ in./ft.})}{(112.5 \text{ in./sec.})(60 \text{ sec./min.})} = 4.27 \text{ minutes per reel.}$$

The number of records in a full reel can be found from the following calculation.

Each record contains	360 fixed characters
and an estimated	128 variables characters
giving a total of	488 characters/record.

Since it is possible to pack up to 12,000 characters per block, the average number of characters will be more than 11,513 characters, i.e.,

12,000	logical upper bound
-487	smallest unit less than one record
11,513	characters/block.

We arbitrarily selected an average packing of 11,750 characters/block. This represents an average packing of 24 records per block. At a storage density of 800 characters/inch, it will require about 14.69 inches of magnetic tape per block. Add to this .75 inches for an interblock gap and the block storage becomes 15.44 inches. Under the read rate of 4.27 minutes/tape, we obtain a rate approximately .095 minutes per 1,000 records, i.e.,

$$(\text{Read rate/record})\ 1000 = \left(\frac{15.44 \text{ inches}}{24.0 \text{ records}}\right)\left(\frac{1 \text{ ft.}}{12 \text{ inches}}\right)\left(\frac{4.27 \text{ min.}}{2400 \text{ ft.}}\right)\left(1000\right)$$

$$= .095 \text{ min.}/1{,}000 \text{ records.}$$

The following table compares the calculation of tape reading time with the time recorded in processing 8 tape reels in response to a request.

NUMBER OF RECORDS IN FILE = 229,000

Calculated read time = (229)(.095) = 21.8 minutes

Observed Processing Time

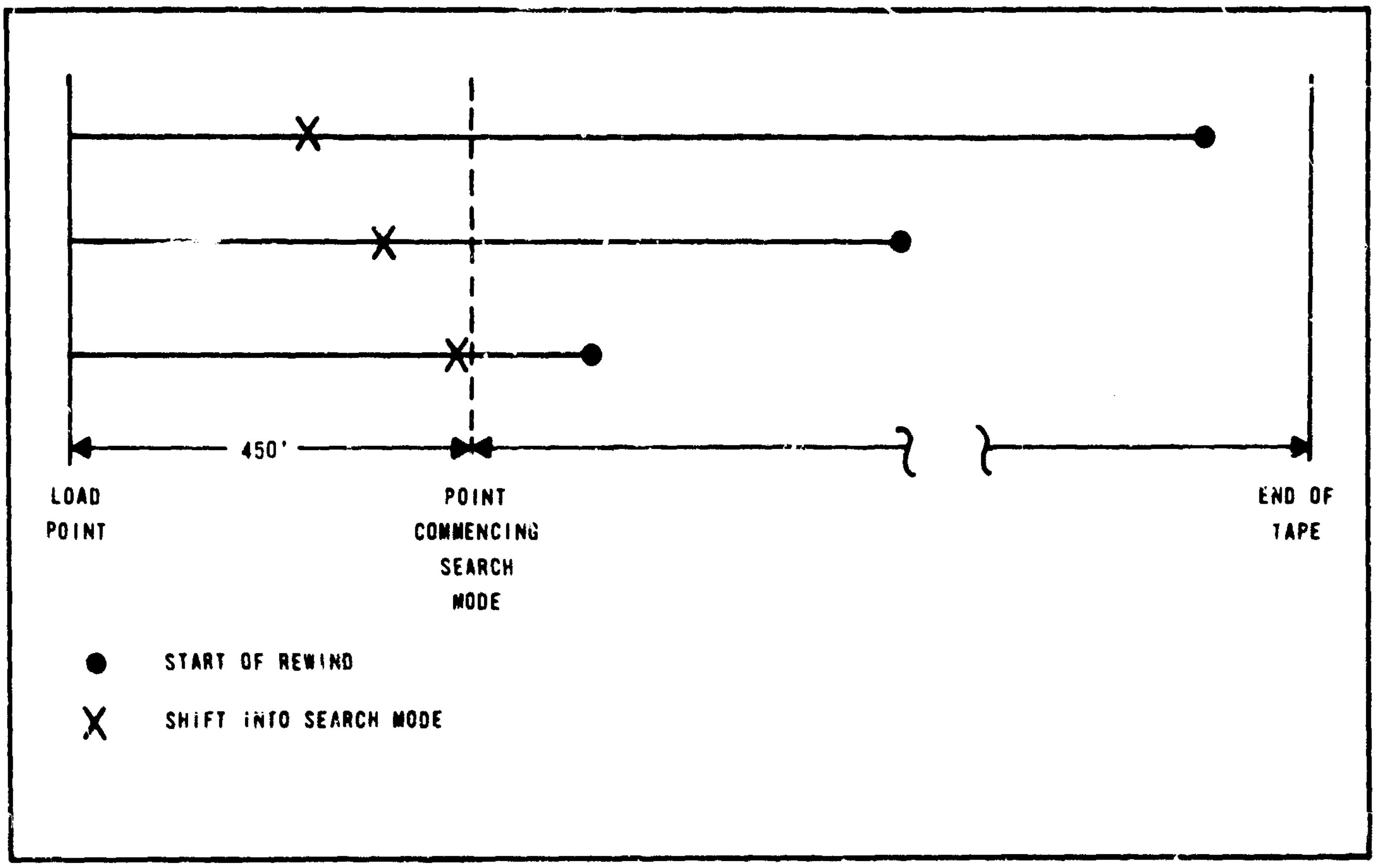
Reel	Read Time
1	4.5 minutes
2	1.0
3	3.5
4	1.5
5	4.5*
6	1.5
7	2.5
8	3.5
	22.5 minutes

*tape stopped several times

Output tape rewind time, T_{RW}, is a function of both the rewind capability of the tape unit and the amount of tape to be rewound. The rewind characteristics of the IBM 729 VI tape unit are

High speed rewind rate	=	500 inches/sec.
Beginning of tape search rate	=	112.5 inches/sec.
Point commencing search	=	450 ft. from load point.

In practice, the high speed momentum will cause the shift into the beginning of the tape search mode to drift past the critical point by a margin that is dependent upon where the rewind phase started, i.e.:



If we consider the maximum rewind to encompass a high speed mode over 2,000 feet of tape with a search speed over the remaining 400 feet, then the maximum rewind time is 90.6 seconds. Thus,

$$0 < T_{RW} < 90.6 \text{ seconds}$$

If the output storage does not pass the point commencing the search mode (about 8,225 records), then

$$0 < T_{RW} < 48.0 \text{ seconds.}$$

Rewind time, during file search, was not considered to be a significant variable; hence was approximated as a constant time expenditure of .5 minute per file search.

File search time, for this sytem, can be reasonably expressed as

$$T_{FS} = (2 \text{ min set-up}) + (.095 \text{ min}) \left(\frac{\text{records in file search}}{1{,}000}\right) + (.5 \text{ min rewind time})$$

or $T_{FS} = 2.5 + (.095) \left(\frac{\text{records}}{1{,}000}\right)$.

The expected file search time for different query types can be determined by estimating the number of records stored on each pertinent tape reel searched. In this example system, one month's history creates approximately

8,000 records in subject area A,

2,000 records in subject area B,

6,000 records in subject area C, and

5,000 records in subject area D.

On the following page is a processing table reflecting the file search time for different depths of searches in the four subject areas. File read time has been rounded to the nearest half-minute. Requests posed against the system may seek data from more than one subject area; thus the estimated file search time is a function of both the depth of the search and the different areas requested. For example, the likelihood of a request searching through months of history from all four areas influences a probability that the file search time will require 17 minutes. This is essentially the methodology used to depict the file search times expressed in the input distribution tables. Two distribution tables were formed; corresponding to a LOW and a HIGH file search effort. The LOW distribution ranges from 3 to 15 minutes -- reflecting requests searching through one month's history from one area of a search through six month's history of all four areas. The HIGH distribution ranged from 15 to 45 minutes -- reflecting a search range from a six-month four-area search to a two-year study of all four areas.

FILE SEARCH TIME = 2.5 MINUTES PLUS

NR MONTH HISTORY	Subject Area			
	A	B	C	D
1	1.0	.5	.5	.5
2	1.5	.5	1.0	1.0
3	2.5	.5	1.5	1.5
4	3.0	1.0	2.5	2.0
5	4.0	1.0	3.0	2.5
6	4.5	1.0	3.5	3.0
7	5.5	1.5	4.0	3.5
8	6.0	1.5	4.5	4.0
9	7.0	1.5	5.0	4.5
10	8.0	2.0	6.0	5.0
11	8.5	2.0	6.5	5.0
12	9.5	2.5	7.0	5.5

D. OUTPUT EXAMINATION

The material in this section highlights some of the capability of the time-flow simulation to support the examination and analysis of a system concept. Specifically, the proposed configuration of the example IR system was simulated and the output (and implications) from this effort is discussed. The analysis methodology employed in this study is essentially of the "guess and test" variety. That is, a proposed concept, stemming from an engineering estimate, is tested through the simulation program; an analysis of the results modifies the concept and the modified concept is tested. This approach would normally continue until the basic concept was accepted or rejected under the performance criteria. In this example study, only the initial test and the first retest are presented.

The output listing (see Appendix B) from the simulation program gives the following kinds of information:

1. Input parameters -- this is a playback of the input data and is provided for convenient reference.
2. Generated query load -- shows the number of each query type posed within each interval.
3. Work load -- depicts the amount of time required at each event to process the queries generated. This output provides a quick picture of the work distribution among the processing functions.
4. Percent use of interlocked events -- illustrates how much of the work time of a component is devoted to different processing tasks.
5. Query processing summary -- gives the number of queries completed, partially processed and not started. This summary also gives the amount of work remaining on the unfinished queries of each query group.
6. Time lost to maintenance -- depicts the time that a component of each event was down for repair over the scheduled availability time.

In addition to these data, there are two major summary listings, i. e., :

7. Summary by query group.
8. Summary by event utilization.

These two categories are discussed in the following description of the simulation study.

1. Simulation Run One

The proposed system concept, query loading factor and work schedule, thus far described in the section, were simulated under the time-flow simulation concept. Figure 6 illustrates some of the data generated in this first run.

	M	T	W	T	F	S	S	M
ACCUMULATED AVERGAE LOAD	10	17	27.6	36.6	42.2			51.8
ACCUMULATED AVERAGE RETURN	2	10.3	16.9	27.5	36.1			43.4
DIFFERENCE	8	6.7	10.7	9.1	6.1			8.4

AVERAGE
PCT EQUIPMENT USE

	M	T	W	T	F	S	S	M	AVE.
SATELLITE A	12.1	17.4	38.5	29.0	11.1			19.4	21.3
SATELLITE B	18.8	26.9	33.4	48.2	34.6			33.7	32.6
CENTRAL PROCESSOR	34.0	47.0	66.1	65.1	24.3			52.9	48.2

DELAY IN QUEUE ATTRIBUTABLE TO

C.P.U.	38.152	TOTAL MINUTES
USER	13.244	TOTAL MINUTES
SATELLITE A	4.555	TOTAL MINUTES
CONSULTANT	3.352	TOTAL MINUTES
SATELLITE B	2.718	TOTAL MINUTES
COURIER	2.134	TOTAL MINUTES

FIG. 6 SIMULATION DATA: RUN ONE

The "accumulated average load" was obtained from the SUMMARY BY QUERY GROUP listing. This summary presents (1) the number of queries within each group arriving within a specified interval of time[1] and (2) the average response time (work plus delay time) for each query group. The table shown on the following page illustrates the average response time (in minutes) for each query type categorized by the day of the week the requests were submitted.

The table illustrates three aspects of the processing effort; i.e.:

1. Some requests generated Friday are held over the weekend.

2. Response time, excluding the requests delayed over the weekend, averages about 24 hours.

3. Delay time accounts for the majority of the response time.

The table does not, however, illustrate where the delay occurs in processing. This will be illustrated in the SUMMARY OF EVENT UTILIZATION.

The SUMMARY OF EVENT UTILIZATION provides the remaining data shown in Figure 6. The "accumulated average load" was obtained by the simple expedient of defining that the user will require exactly one minute to accept the IR output. This function is labeled EVENT 14 in the simulation; thus a use time of two minutes for EVENT 14 on Monday indicates that two system outputs were received on Monday. A comparison of the "accumulated average load" with the "accumulated average return" reveals that the system is completing yesterday's work today; i.e., by Monday, an average of 43.4 requests were completed and returned to the users -- on Friday, a total of 42.4 requests had been accumulated. This indicates that a processing backlog is <u>not</u> building up.

Event utilization is summarized by depicting (1) the average amount of time each service unit of each event is utilized[2], (2) the percentage of the scheduled time reflected in this usage, and (3) the average delay time accumulated in queue before each event.[3]

---

[1] The output intervals are specified by the investigating engineer. In this study, each work day was designated as an output interval; thus the summary depicts the daily "history" of the system concept being simulated.

[2] An "NQ" listed under the service unit number indicates that the event simultaneously processes all data in queue when the event becomes available.

[3] Delay time in queue is the sum of all the times that data are delayed; thus 2 elements waiting 3 minutes = 6 minutes delay time.

QUERY GROUP	MON	TUE	WED	THU	FRI		MON	AVERAGE RESPONSE TIME	AVERAGE DELAY TIME
11	948 min.	1633	1100	1549	4040		496*	1466	1394
12	1339	1724	1684	2268	0		395*	1813	1662
22	874	1009	1492	761	178		354*	1062	846
23	946	1330	1348	1075	4110		203*	1813	1720
31	1315	1277	1797	756	0		380*	1325	1079
33	996	1178	975	1030	4174		153*	1784	1695

response time slightly under three days indicating that some requests are held over the weekend.

* queries not completed.

NOTE: 1440 minutes = 24 hours.

Taken together, this output illustrates equipment/personnel utilization, processing bottlenecks and points where delays occur in processing. Figure 6 shows that the major processing delay in the proposed design occurs at the CPU and at the user.

The "average percentage of equipment utilization", however, indicates that the delay at the CPU is not caused by an insufficient amount of scheduled time. This indicates that the problem is connected with when the computer is available, instead of how often it is available. Similarly, the queue formed in front of the user is created by data being returned during evenings or nighttime when the user is unavailable to receive the output. Much of this is probably caused by data delaying over the weekend.

The following table illustrates the day-by-day utilization of the CPU. Since the central processor is scheduled between 0700 and 1400 each day, the delay in queue represents approximately 4 queries waiting to be processed over the 17-hour interval that the CPU is not available. The 20,860 delay in queue on Monday represents about 5 queries being delayed over the weekend.

USE OF CENTRAL PROCESSING UNIT

	AVERAGE WORK TIME	% SCHEDULED TIME USED	DELAY IN QUEUE
MON	143	34%	43
TUE	198	47%	4,744
WED	278	66%	4,368
THU	273	65%	4,564
FRI	102	24%	3,573
///			
MON	222	53%	20,860
TOTALS	1,217	48	38,152

Examination of the "average percentage utilization" of the scheduled time revealed that no service unit was utilized more than 50% of the available time.[1]

2. Simulation Run Two

The analysis of simulation run one was essentially that (1) the system response time was about 24 hours, (2) the processing schedule was not particularly efficient and (3) there are an adequate number of components available to accomplish the generated work load. Based on this analysis, the following conjecture was postulated:

a. The processing facility schedule could be shifted to the second shift (between 1700 and 0800) without significantly effecting system response time.

b. The scheduled service unit availability time could be reduced without effecting system response time.

This conjecture was tested by altering the inputs into the program and running a second simulation.

Figure 7 illustrates the changes in the second input. For this new schedule, the user still works between 0800-1200 and 1300-1700 Monday through Friday. The system consultant (EVENT 30) has shifted his schedule so that he is available between 1300 and 2100 each work day. All basic system components have been rescheduled under the night shift, i.e.,

between 2400 and 0800 MON

between 1700 and 0800 MON-THU

between 1700 and 2400 FRI.

[1]It should be noted that the percentage use of an NQ event has little meaning. The percentile is calculated by dividing the work time by the available time. An NQ event will simultaneously process all data in queue when the event becomes available. The amount of "work" expected is the sum of all the work in queue.

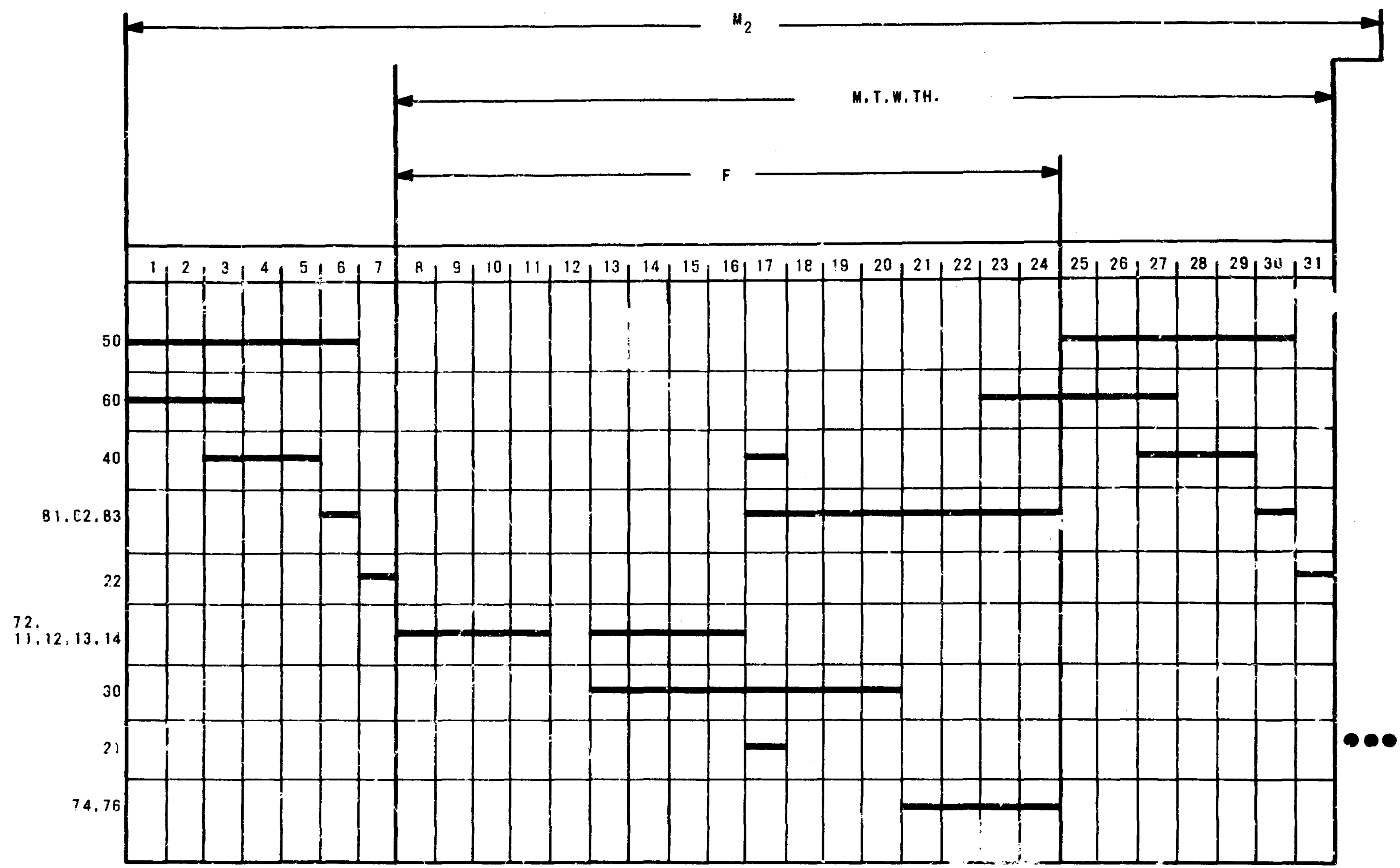


FIG. 7 PROCESSING SCHEDULE TWO

Additionally, the availability time of some of the components has been reduced, i.e.:

CPU	from 7 hours to 5 hours daily
COMPUTER A	from 7 hours to 4 hours daily
COMPUTER B	from 16 hours to 6 hours daily

All other simulation parameters remained the same under the second run.

Figure 8 illustrates the second simulation output and Figure 9 compares this output with the previous simulation response. There are several aspects of this comparison that should be noted; i.e.;

1. The generated "average daily loads," while similar, were not identical. Since the load factor for both simulations was the same, it is believed that the difference can be attributed to the fact that only three iterations of each simulation were run. A higher number of iterations would tend to dampen differences in the generation of random numbers used in the program.
2. The "average response time" did change under the modification of the system concept.
3. The "average percentage use of equipment" was higher under the new schedule than under the previous work schedule.

An examination of the output listed in the SUMMARY BY QUERY GROUP (see Figure 10) revealed that the major contribution to the increase in response time occurred on Thursday for query groups 11, 12, 31 and 33. Query group 22 picked up a significant increase on Monday and over the weekend. Query group 23 was not particularly affected by the change in schedule. The shift in schedule to nighttime processing establishes a base line of 15 hours for the minimum response time for queries submitted Monday through Thursday. Queries submitted Friday cannot be returned until at least Monday. Under the prior schedule, it was possible to receive output the same day that a query was submitted. This fact predominately accounts for the increased response time for query group 22; it does not, however, explain the major increase on Thursday.

	M	T	W	T	F	S	S	M
ACCUMULATED AVERAGE LOAD	8.7	14.4	24.1	34.1	39.8			48.8
ACCUMULATED AVERAGE RETURN		7.3	13.6	20.9	29.5			45.5
DIFFERENCE	8.7	7.1	10.5	13.2	10.3			3.3

AVERAGE PCT EQUIPMENT USE

	M	T	W	T	F	S	S	M	AVE.
SATELLITE A	35.8	45.4	44.1	41.8	23.8			18.6	31.9
SATELLITE B	66.2	58.1	99.2	100.0				83.7	81.7
CENTRAL PROCESSOR	66.0	59.1	80.3	82.8	95.5			88.5	70.2

DELAY IN QUEUE ATTRIBUTABLE TO

SATELLITE B	37,319	TOTAL MINUTES
C.P.U.	18,495	TOTAL MINUTES
USER	9,041	TOTAL MINUTES
CONSULTANT	6,500	TOTAL MINUTES
COURIER	3,477	TOTAL MINUTES
SATELLITE A	1,544	TOTAL MINUTES

FIG. 8 SIMULATION DATA: RUN TWO

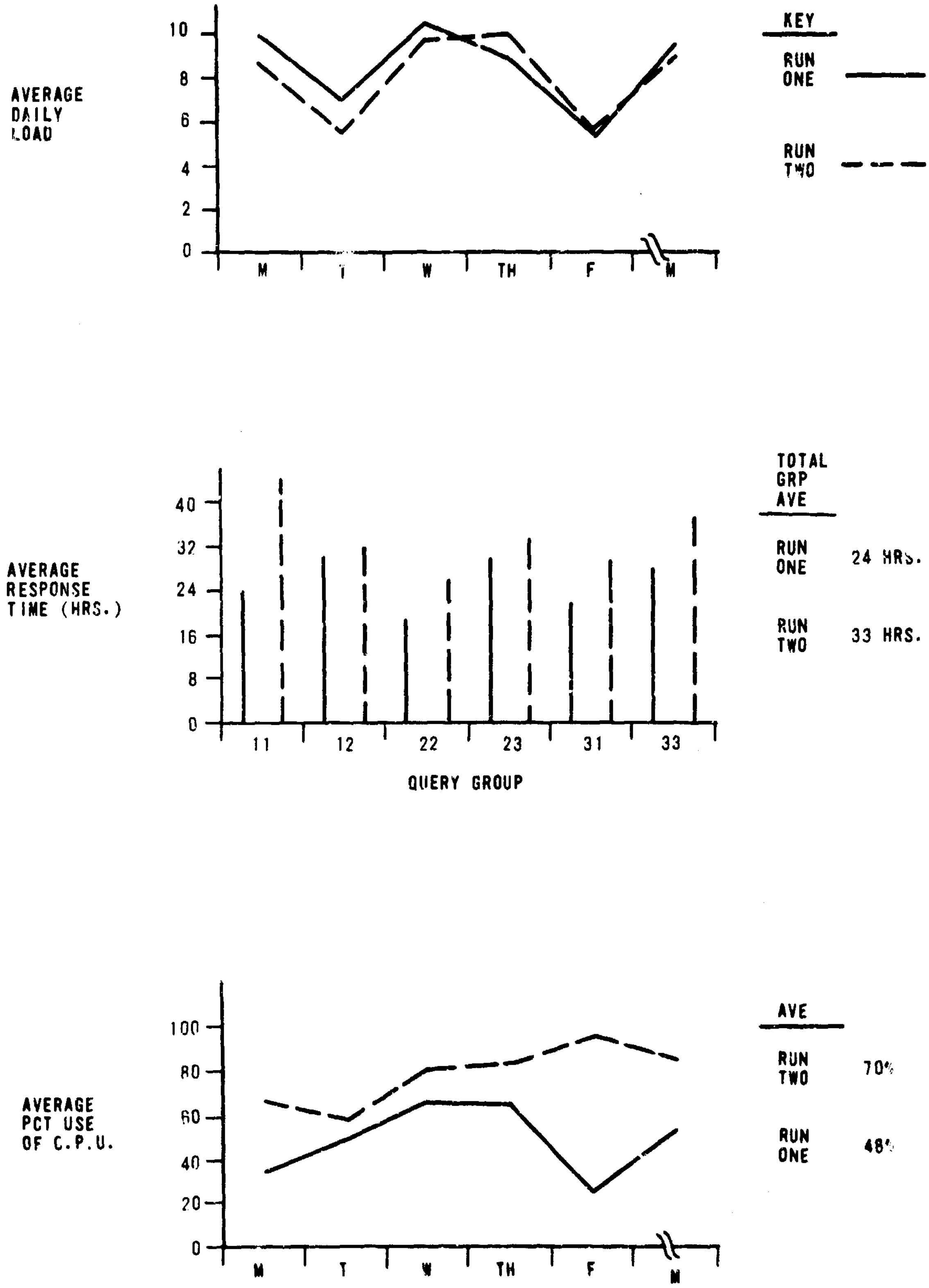


FIG. 9 COMPARISON OF RESPONSES FOR THE EXAMPLE SYSTEM

	MON	TUES	WED	THU	FRI	MON	AVG RT	AVG DELAY
11 RUN 1 RUN 2	948 1378	1633 1283	1100 1589	1549 3567	4040 5502	496* 1305	1466 2652	1394 2570
12	1339 1328	1724 1089	1684 1461	2268 4062	0 0	395* 1365	1813 1912	1662 1773
22	874 1895	1009 1218	1492 1808	761 1220	178 5669	354* 1257	1062 1606	847 1400
23	946 1180	1330 1166	1348 1154	1075 1953	4110 4357	203* 1111	1813 2106	1720 2029
31	1315 1311	1277 2067	1797 2692	756 1517	0 0	380* 1092	1325 1850	1079 1617
33	996 1676	1178 1521	957 1416	1030 4423	4174 4043	153* 1021	1784 2378	1695 2289

○ response time about doubled under the second schedule.

* queries not completed.

NOTE: 1440 minutes = 24 hours.

FIG. 10 COMPARISON OF RESPONSE TIME UNDER THE TWO PROCESSING SCHEDULES

The following table illustrates the utilization of the CPU and the two satellite computers under the second processing schedule.

	% SCHEDULED TIME USED			DELAY IN QUEUE		
	CPU	A	B	CPU	A	B
MON	66%	36%	66%	2,540	297	482
TUE	59	45	58	1,754	233	807
WED	80	44	99	2,728	308	1,138
THU	83	42	100	3,394	393	3,617
FRI	96	24	NOT SCHEDULED	1,858	236	
///						
MON	89	19	84	3,262	76	2,653

Satellite computer A (performing paper tape-to-card conversion and off-line printing) is not being fully used during the scheduled time. Satellite B (performing off-line and remote printing) becomes saturated on Wednesday and Thursday. This performance follows the average daily load (see Figure 9) which increases to a peak on Wednesday and Thursday.

The conjecture that both (1) the processing schedule could be shifted and (2) component availability could be reduced, was not entirely true. It would seem, however, that the availability of the components could be adjusted to better conform to the daily expected work load. Moreover, an adjustment of this nature would both (1) more efficiently schedule the availability of the system components for IR (thereby freeing them for other processing efforts elsewhere) and (2) eliminate the Thursday bottleneck. This conjecture has not been tested.

The question of whether this second conjecture would produce a good design (or, for that matter, if any of the proposed configurations are acceptable) cannot be answered directly in the simulation. System acceptability is a function of how well a concept satisfies the performance criteria established for the operation. The time-flow simulation does, however, provide a vehicle for examining the effects of a processing concept under conditions that are significant to the evaluation of the systems.

Reverse (Page 58) Blank

IV. RECOMMENDATIONS

This research effort has examined the use of simulation as a technique for analyzing and evaluating information storage and retrieval systems. While the techniques and programs discussed in this report have not been fully tested, there is reason to believe that the concept is feasible and should be developed to provide engineers and manager with an analytical tool for systems planning and evaluation.[1]

Development of the generalized simulation model, however, should include both modifications and additions to the present simulation structure. The recommendation for development, therefore, includes suggestions for the refinement and expansion of the simulation concepts as presented in this report.

A. SIMULATION REFINEMENT

The simulation structure developed in this effort was not specifically engineered to facilitate application; but instead, was put together to expedite program testing and concept examination. The present numerical language and format are somewhat complex for efficient expression of a system's operational characteristics. Experimentation with the simulator has revealed that the simulation usage could be enhanced by refining the simulation language. It is recommended that --

1. a mnemonic language with an open structure be developed in lieu of the numerical expression now used to denote processing at each step;
2. event availability be defined in terms of the different processing events, not in terms of homogeneous sections of available processing time;
3. the engineer be allowed to communicate with the output program to express his needs for analysis data.

---

[1] This belief stems partially from our own study and test application; partially from the prior works of McKenney and Allen at Harvard ("A Computer Center Simulation Model"); and partially from the program efforts at IBM ("General Purpose Systems Simulator III, Introduction," B20-0001-0).

The introduction of words as a basis for a tailored language to express systems configurations and operations would seem to provide an answer to the language problem. A few simple rules of syntax and the use of key words, concepts similar to those employed in the COBOL language, can supply a very simple, but expressive means of conveying all the desired systems information to the simulation program. In addition, it is much easier to modify or expand such a language when other types of capabilities or specialities are to be added to the simulation.

The query description currently employed, a combination of numeric codes in a fixed sequence[1], can be converted to English language statements in combination with certain key and optional words which provide smoother reading.[2] For example, typical statements could be --

THREE: USE KEYPUNCH AND OPERATOR WITH TIMING
FROM DISTRIBUTION 5 SCALED BY 2.3.
GO TO FOUR OR FIVE DEPENDING ON PROBABILITY 3.

---

1 The reader is directed to APPENDIX A, SECTION B, INPUT PROGRAM for a complete explanation of the current model's numeric language.

2 Employing the connotations of language explanation utilized in presenting the COBOL or PL-1 languages, the concept of a tailored language can readily be presented. Realizing that --

a. script letter words indicate locations where the program USER inserts his own desired words or phrases;

b. capitalized letter words underlined indicate key words which must appear precisely as shown;

c. capitalized letter words indicate optional words which need not appear, but if they do, must appear precisely as shown;

d. brackets ([ ]) indicate optional additions;

e. braces ({ }) indicate that there is a choice of items that must appear at the particular location.

A typical series of query description statements could be --

[label:] USE event [AND event [AND .....]] WITH TIMING

FROM { a constant / a distribution name / CONTENT OF a variable } [AND { } [AND.....]]

[SCALED BY { a constant / a distribution / CONTENT OF a variable } [AND { } [AND.....]].

[label:] GO TO label [OR label [OR label.....]]

DEPENDING ON { a constant / a distribution name / CONTENT of a variable } .

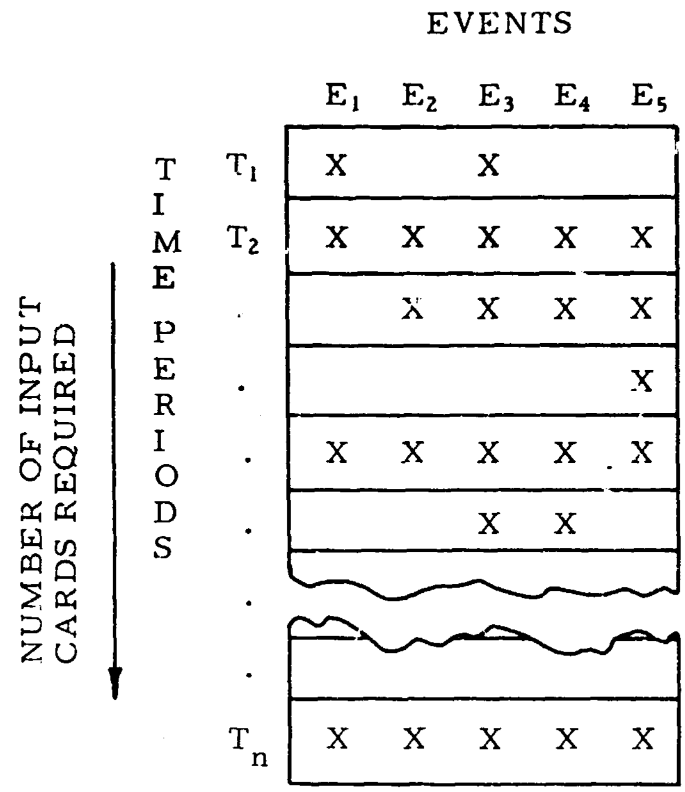
Such statement formats enable the engineer to --

1. readily identify the particular statement;
2. enter the name of an event instead of a numeric code;
3. indicate the need for simultaneous processing by two or more events;
4. utilize conventional use time sources and multiplicative factors or an indirect addressing capability;[1] and
5. indicate a path of processing flow dependent upon any number of defined strategies.

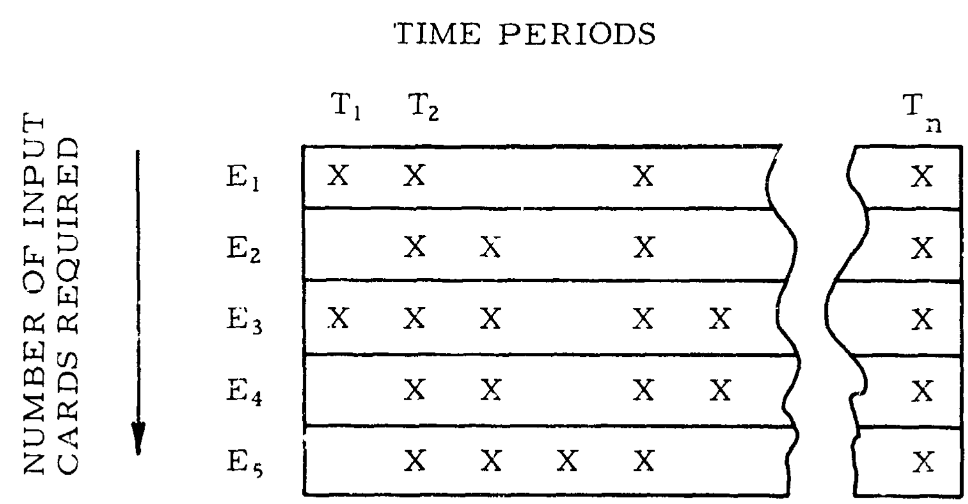
Other statements of similar construction and method of assemblage can be employed to indicate special processing conditions.

The simulation language development would also provide the ability to reduce the amount of necessary input entry required of the engineer. For example, the operating schedule currently requires that all events available during a homogeneous time period be specified. Consequently, if event operations do not coincide over many sequential time periods, a large number of scheduling cards must be entered as diagrammed --

[1] Indirect addressing allows the appropriate selection from one of several time distributions which could represent, for example, high, medium, or low operating time for a particular event or job. Another usage of indirect addressing is to preserve the dependency relationship among some events by providing a conditional relationship in the selection of operating time.



However, indicating the operating time periods by event should drastically reduce the number of cards entered since scheduling is now independent of time periods as diagrammed --



Another type of statement could be employed to ask the program for additional output information; information other than a fixed minimum output for the program. Thus, if the engineer wants to know the number of errors at some defined decision point, he can specify that a tally be made of the errors generated at that point.

B. SIMULATION EXPANSION

Some existing information systems employ strategies of operation not incorporated in this simulation model. Therefore, the prime recommendation of this report is that any development efforts considered include the removal of basic and fundamental restrictions of the model which may limit its real-world approximating capability. Paramount of these limitations are the methods for assigning the queries for processing and for scheduling components of the system.

Alternate operational strategies could be presented to the model as strategy modules containing algorithms of significant real-world behaviors. Therefore, it is specifically recommended that different modules be developed which could be added to or deleted from the model. Such an approach provides the systems engineer with various alternative strategies for operating the particular system.

Initial feasibility studies strongly indicate that the majority of the types of operational strategies employed today can be readily incorporated into the model. The recommendations of specific strategies for development are:

1. Additional scheduling capabilities to include input processing, systems updating and production processing.
2. Procedures for specifying alternate methods of assigning queries for processing such as priority interrupt, length of queue versus required processing time allocations, boundary processing continuation conditions in time, and variable man-machine matching.

1. Scheduling

The current model assumes the data base condition for any retrieval effort is ideal; i. e., it is always current and complete. The updating of the current model's data base with the latest information addition and redundancy or invalidity removal can only be approximated by initiating a specially defined query type during some Δt. But the query processing would be subject to the same processing criteria as any other query type. In order to provide a better approximation of real-world scheduling, information retrieval systems support file maintenance simulation should be improved.

File maintenance includes the entire process from encoding to entering the data into the data base storage. Some queries are specially delayed to insure the availability of the most recent information. In addition, normal query processing is also delayed. One simulating strategy which could be employed is the "restricted schedule." This procedure would essentially close a defined section of the system for processing except to particular operations such as updating flow. Such a strategy could be initiated at designated periods of time.

The restricted schedule concept can also be utilized to simulate production processing. Information systems today are the initiating point for numerous reports ranging from daily current status to monthly summaries in content. In order to provide these services of particular outputs, a rigid schedule must be maintained. Therefore, initiating defined report processing with the restricted schedule priority will demonstrate the influence of assembling such reports on the operations profile of a system.

A third type of scheduling expansion is the revamping of the strategy for simulating event failure and/or event maintenance. Real-world probable breakdown or preventative maintenance scheduling is normally based upon the lapsed time since the last maintenance overhaul and the amount of time that event has been used in the meantime. There may be other timing strategies applicable to specific events.

2. Assignment

Expanding the systems environment monitoring capability will provide the ability to compare the state of the system against a list of *a priori* rules or strategies governing processing assignments thereby increasing the operating decision making capability. There are numerous assignment strategies, some very practical, some highly theoretical.[1] However, including several allocation strategies in addition to the first-come-first-served basis now employed will definitely enhance the utility of the model.

[1] Denning, P. J., "Queueing Models for File Memory Operation," MIT Project MAC, MAC-TR-21 (Thesis), October 1965.

One such allocation strategy is priority interrupt. By defining an ordered list for queue location of types of requests, each query arriving for processing at an event can be properly pigeonholed. Assignment out of queue from any ordered group would still be on a first-come-first-served basis. Thus, if an event is processing priority 2 class request, a newly arrived priority 1 request would be entered as the next job to be processed by the event. Had the event been processing a series of priority 1 requests, a newly arrived priority 1 query would have become the last member in the priority 1 queue.

Other allocation strategies employ more detailed heuristic searches. An example would be alternate routing during processing.[1] Using this type of assignment, the selection of an event to process a query could be based upon a comparison among (1) the length of queue at the required event, (2) the length of queue at possible alternate events used for the same type of processing, (3) the length of queue at other events in the query's processing path which do not require any previous processing sequence, and (4) the required event processing time for the query. Such comparisons provide the basis for optimum processing assignment. Another similarly involved allocation is Round Robin Scheduling.[2] Other assignment variations might include Batch Processing where a certain number of like requests are collected and then processed simultaneously.

Any processing of a query by an event requires some "fixed" length of time for completion. These fixed lengths can extend beyond certain time boundaries of operation. A boundary may be defined by lunch time or quitting time or by component operation in terms of the systems operating schedule. Therefore, a strategy for optimum processing time could provide a criteria for event shutdown involving a partially processed query by position in time. Such a decision of whether to completely process work beyond a scheduled closing of a component in the system or even begin its processing could be decided after weighing the importance (or priority), length of processing time required, and the boundary point in time (a morning coffee break versus quitting time).

[1] Russo, Francis John, "A Heuristic Approach to Alternative Routing in a Job Shop," MIT Project MAC, MAC-TR-19 (Thesis), June 1965.

[2] Greenberger, Martin, "The Priority Problem," MIT Project MAC, MAC-TR-22, November 1965.

The assignment of an event for processing is insufficient in many cases to insure the actual processing. For example, assigning data to a key punch operation without an available operator does not produce punched cards. The current model assumes ideal personnel allocation. Whenever there is an event required for processing, any necessary operator is also instantaneously available. In order to more realistically pattern real-world operations, a module for assigning variable operator availability could be developed. Then the number of operators available could be varied over time, and actual query processing would be a function of operator availability as well as event availability. Pooling strategies can be developed and evaluated and personnel with several assigned areas of responsibility can be provided an ordered priority listing for assignment.[1]

---

[1] McKenney, James L., and Allen, B. L., "A Study of a Man-Model Symbiosis Controlling a Computer Center," unpublished paper.

APPENDIX A

THE SIMULATION PROGRAM DESCRIPTION

APPENDIX A   TABLE OF CONTENTS

The simulation output is achieved after the input parameters and constraints have been processed by an iterative procedure through the distinct subroutines of the computer program. This iterative procedure continues until (1) the predetermined number of iterations have been completed or (2) the operator stops the program.[1] The subroutines function as a chained sequence of logical processing steps.[2] Such chaining or segmenting provides the computer simulating program with an ability to efficiently utilize small ADP systems that have a random access capability to secondary bulk storage. The present program is of such size, however, that a similar chaining approach would probably be applicable even for larger ADP systems.

The model program, written in FORTRAN II, has been developed for one of the ADP systems available in the research facilities at HRB-Singer, Inc. The particular configuration utilized was selected because it was convenient and because it was thought to represent a fairly popular and, therefore, readily available ADP system. This system consists of --

(1) an IBM 1622 Card Read Punch,

(2) an IBM 1620 MOD II Control Processing Unit,

(3) two IBM 1311 Disk Storage Drives, and

(4) an IBM 1443 Printer.

The current model programming employs eleven subroutines which can be appropriately grouped as illustrated in Figure 11. The function and particular aspects of each of these groups is presented in the following sections, although not in their proper program order.

---

[1]The mean and variance of the work load are calculated as an aid in determining when a representative sample has been approximated.

[2]The program utilizes the IBM I/O macro-statement CALL with the operand LINK to achieve the linkages.

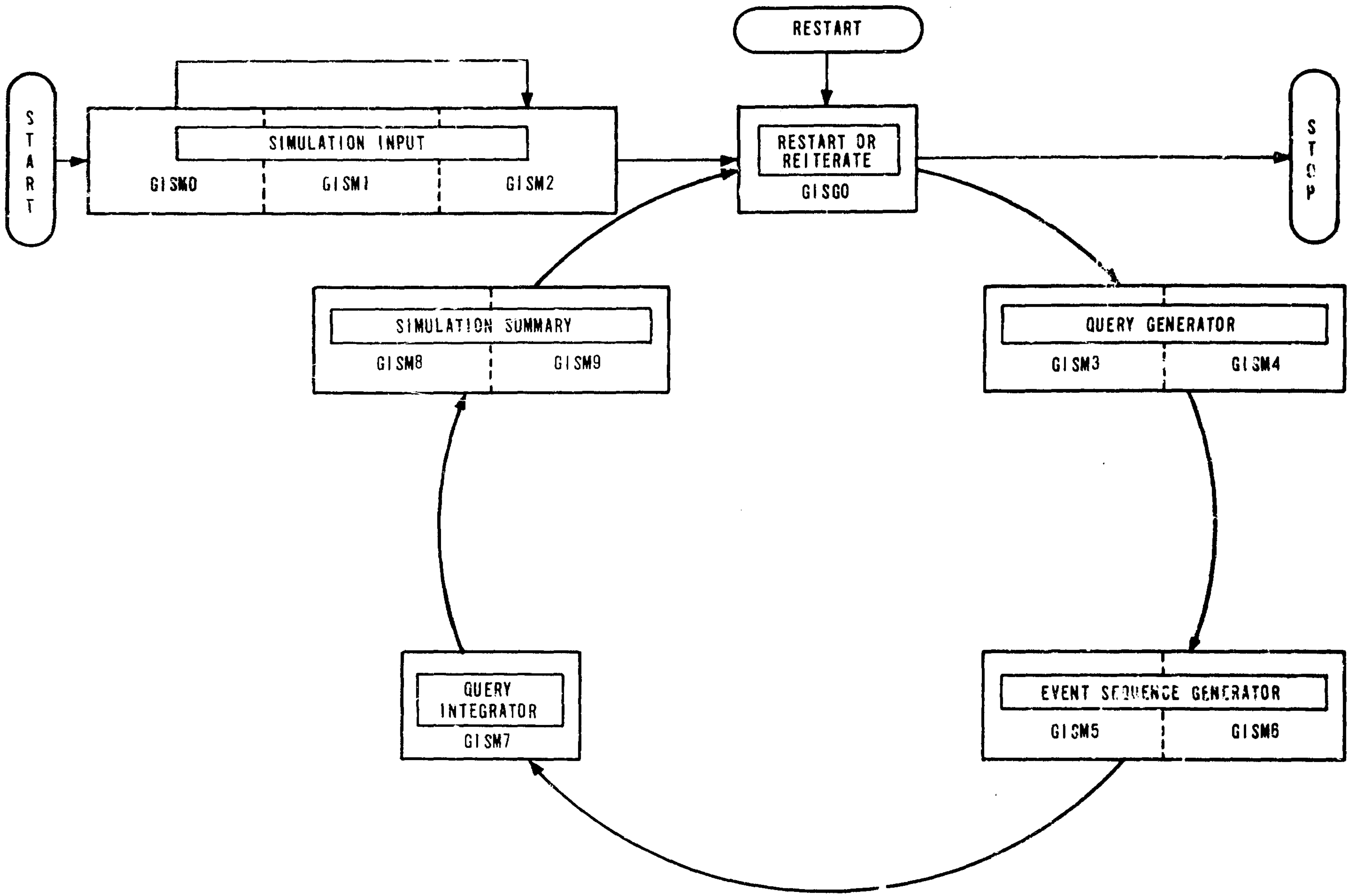


FIG. 11 ORGANIZATION OF THE SIMULATION PROGRAMS

A. SUMMARY PROGRAM

The summary program analyzes and assembles the majority of the developed data into an output presentation. The information developed during each iteration can be highly significant particularly when indicating extreme processing conditions. But equally revealing is the average processing profile of the system produced by accumulating the data developed over all iterations. Therefore, the output values are appropriately presented not only for the particular iteration, but also as accumulated over all completed iterations. The output is presently printed after every iteration. However, it is possible to modify the program so that the operator may optionally specify which iterations are to be printed.

The various output data can be subdivided into three general areas --

(1) a summary of query processing and component failure;

(2) a summary of the average response time for each query group; and

(3) a summary of event utilization.

The output data for the average response times and event utilization are summarized within and over all the time intervals that have been designated by the systems engineer over the time line. These designated output intervals allow the systems engineer to examine processing within particular sections of the time line. The output intervals are defined by integral multiples of Δt and may vary in number from 1 up to a maximum of 25. Although these output intervals may not overlap, they may represent different time spans and may be separated. However, if the time line is not totally partitioned into output intervals, any information contained in the undefined time spans will not be included in the summary for the time line. An example of the simulation output is presented in Appendix B of this report.

1. Query Processing Summary and Time Lost Due to Maintenance.

The distribution of all queries initiated during the simulation is tabulated by

a) the number of queries that were completely processed during the simulated time span;

b) the number of queries that were partially processed during the simulated time span; and

c) the number of queries that were initiated but not processed during the simulated time span.

A breakdown for the partially and nonprocessed queries is calculated to indicate the total amount of "work" time remaining. The number and work time for these queries is also shown as it is distributed among the different query groups.

The event availability time lost through component failure or equipment maintenance is calculated for every event. The time lost for each iteration is denoted by the number of Δt time segments affected.[1] Therefore, this number times the time span of a Δt gives the amount of scheduled time lost. The accumulated time lost over all completed iterations and the average time lost per iteration is given in actual time units (e. g., minutes).

2. Query Group Response Time

The response time or processing time for a query is defined as the time the query is actually being worked on plus the time the query is being delayed. The response time for a query group is the average response time for all the query types in the query group.

The number of queries that were initiated during each output time interval is tallied for each query group and is printed with the query group's average processing, working and delay times. Those groups that contain queries that were not completed are appropriately designated. Response time for each query group is also averaged over all output intervals.

[1]In the present simulation, one service unit of an event is down an entire Δt time interval if the random number test indicates that the event has a failure. This test is made at each Δt against the maintenance probability assigned by the investigating engineer.

3. Event Utilization

The utilization of service units for query processing is calculated for each output time interval and then summarized over all of the intervals. For each output time interval, the use of each service unit is calculated and is presented with the percentage of the available time the service unit was used. For example, assume that an output time interval is 500 units in length and that a service unit was scheduled for an event throughout the entire output time interval; then if the service unit used only 250 time units, the 250 figure and 50% usage would be printed. However, if during the time span of 500 units the service unit was only scheduled for 250 units and it was used for all 250 units, the 250 figure and a 100% usage would be shown. Additionally, the total delay time accumulated against each event (and therefore, all service units within the event) is calculated. [1]

[1] Delay time is the sum of the waiting time of all units in queue before an event. Thus, two units in queue for 3 and 4 minutes, respectively, each have a delay factor of 7 units.

B. INPUT PROGRAM

The input routine consists of three subroutines which act as a compiler and assembler for a special language developed for this simulation program. The input parameters and constraints are presented to the simulator on cards in a somewhat simple numerical language using prescribed formats.[1] Essentially this routine identifies the card type, reads the card's content, verifies to some extent the completeness and accuracy of the input data and prints the program's interpretation of the data for the operator's or the engineer's convenience.[2]

Before preparing the input parameter and constraint cards for the simulation, preliminary assemblage of the required information is advisable. Pictorial representation of the system, listing of the events, charts indicating event availability schedules, as well as flow charts and forms for developing and documenting the query type description are very useful in designating and developing proper input information. The following material briefly describes each input card used to load data into the simulation program.

---

[1]This approach is similar to IBM's AUTOCHART or GPSS (General Purpose System Simulation) languages.

[2]The alpha-numeric codes used to identify the card types also provide the ability to properly sort the entire INPUT DATA PACK prior to starting the simulation program.

1. Card Type No. 1 -- IDENTIFICATION

This input card enables the engineer to label the particular simulation (and its associated input and output data) by allowing a 70-character description to be printed on all pages of the output listings. There is only one identification card, but it must always be present (even if blank) as the first data card when reading input.

The required card format is --

Col 1-2	'ID'
Col 3-72	Any 70 characters to be used as a label on output.
Col 73-80	not used

2. Card Type No. 2 -- CONTROL

This input card controls the data input procedure through letter indicators denoting subsequent instructions for reading particular sets of input data. This control permits all or any part of the parameters to be changed during successive simulation runs. The input data deck could conceivably approach a maximum of about 1500 cards. The control card must always appear as the second input card and must appear every time data are entered.

The required card format is --

Col 1	'K'
Col 2	"L" if list of events codes are to be read, otherwise blank
Col 3	"M" if meaning of query codes are to be read, otherwise blank
Col 4	"P" if probability distributions are to be read, otherwise blank
Col 5	"Q" if query descriptions are to be read, otherwise blank
Col 6	"R" if arrival of queries are to be read, otherwise blank
Col 7	"S" if schedule of operations are to be read, otherwise blank
Col 8-80	not used

3. Card Type No. 3 -- LIST OF EVENT CODES

These input cards allow the systems engineer to define up to 35 events that are characteristic of the system being simulated. The simulation recognizes and can appropriately handle three types of events:

1. NORMAL EVENTS -- service units process one element at a time. The delay time in queue for a normal event is a function of both the scheduled availability of the event and the number of elements waiting to be processed. A card punching operation is an example of a normal event.

2. NQ EVENTS -- service unit will begin to simultaneously process all items in queue when the event becomes available. Delay time in queue is a function of scheduled event availability only. A courier pickup is an example of an NQ event.

3. INTERLOCKED EVENTS -- service unit is capable of performing several different functions (though it can perform only one task at a time). A satellite computer that drives both a printer and a card-to-tape conversion operation interlocks these two events if only one function can be performed at a time.

Each particular event is assigned a numeric code (a positive integer of one or two digits excluding the numbers 98 and 99). Additionally, the number of service units available in each event must be defined. The number of service units assigned to each event need not be the same, but the total service units for the system being simulated cannot exceed 100. Since certain events may have an expected maintenance or failure profile, a probability of an event having a failure in any Δt interval can also be designated. When such an unscheduled shut down occurs, the 1st service unit is removed for a Δt time.

The required card format is --

Col 1	'L'
Col 2-3	Event Code number
Col 4-5	not used
Col 6-65	Any 60 characters describing the event
Col 66-67	"NQ" if the event is to be considered this type

Col 68-70 Number of service units for this event

Col 71-73 not used

Col 74-75 Percent probability of failure for maintenance

Col 76-78 not used

Col 79-80 Event with which this event is interlocked

Notes: 1 -- The last 'L' card must contain L99 in Cols 1-3 if fewer than 35 cards are to be used.

2 -- A maximum of 35 cards may be used.

4. Card Type No. 4 -- MEANING OF QUERY CODE

These input cards allow the engineer to identify up to 200 query types which may be distributed among a maximum of 49 query groups. The query type and group is identified by a four-digit positive integer where the first two digits define the query type and the second two digits define the query group. For example, the numbers, 101, 102, 103, 104, 105 would indicate 5 query types for one query group where the numbers, 101, 201, 301, 401, 501 would indicate one query type for each of five query groups. There is a maximum of 99 query types for each query group.

The required card format is --

Col 1	'M'
Col 2-3	Query group number
Col 4-5	Query number within the group
Col 6-65	Any 60 characters describing the query
Col 66-80	not used

Notes: 1 -- The last 'M' card must contain M9999 in Cols 1-5 if fewer than 200 cards are to be read.

2 -- A maximum of 200 cards may be used.

3 -- A maximum of 49 query groups is allowed.

5. Card Type No. 5 -- INPUT PARAMETERS

This input card defines the number of Δt's (input time intervals) used by the systems engineer to subdivide the simulation time span. It also identifies the basic time unit of the simulation and defines the time span of a Δt in these basic time units. This card must be included whenever any data is read into the program. Up to 400 input Δt's can be designated.

The required card format is --

Col 1	'N'
Col 2-7	not used
Col 8-10	Number of time sectors in system input description
Col 11-12	not used
Col 13-20	Number of units per time sector
Col 21-30	10 characters describing time unit (seconds, minutes, hours, days, etc.)
Col 31-80	not used

6. Card Type No. 6 -- OUTPUT PARAMETERS

These two input cards define the lengths of the output time intervals (which are some multiple of a Δt) by defining the beginning Δt and the ending Δt encompassing each output interval. Both of these cards must be included when reading data into the program. The number of output time intervals may not exceed 25. Although the output sectors may not overlap, they may be gapped and they may represent different spans of time.

The required card format is --

Col 1	'O' (letter O, not zero)
Col 2	'1' or '2'
Col 3-5	Time interval # at which first output time sector begins
Col 6-8	Time interval # at which first output time sector ends
Col 9-11	Time interval # at which second output time sector begins
Col 12-14	Time interval # at which second output time sector ends
Col 15-17	Time interval # at which third output time sector begins
Col 18-20	Time interval # at which third output time sector ends

etc.

13 output time sectors described thus on card 01

12 output time sectors described thus on card 02 (Cols 3-74)

Notes: 1 -- Both cards must be present.

2 -- For less than 25 output time sectors leave remaining columns blank.

3 -- Output time sectors may not overlap but may be gapped.

7. Card Type No. 7 -- PROBABILITY DESCRIPTION

A basic contention of this model is that the time utilized in an event can be expressed in an accumulative histogram. These input cards simplify the defining of a time distribution by providing cells for 20 integers which represent the time at every 5% probability in the ogive. Up to 49 ogives can be loaded into the simulation. Each ogive is identified by a designated numeric code up to two digits in length.

The required card format is --

Col 1	'P'
Col 2-3	Distribution identification number
Col 4	Card number (1 or 2)
Col 5	not used
Col 6-10	Time used at 5% (or 55%) probability (according to card #1 or 2)
Col 11-15	Time used at 10% (or 60%) probability
Col 16-20	Time used at 15% (or 65%) probability
	|
Col 51-55	Time used at 50% (or 100%) probability
Col 56-80	not used

Notes: 1 -- Last Probability Distribution Card must contain P99 in Cols 1-3 to signal end of such cards if fewer than 49 distributions are to be read.

2 -- A maximum of 49 distributions (98 cards) are allowed.

Card Type No. 7A -- PROBABILITY MULTIPLICATIVE CONSTANTS

These two input cards allow the engineer to designate a floating point value that can be used to multiply the quantity selected from a time distribution. This capability provides for situations when

1) two events have the same distribution range but the actual time values differ by some proportional factor, or

2) it is not possible or desirable to express the time units in basic time units.

A problem can develop when using these factors since the particular computer system used for the development of this simulation program will not handle integer values greater than 4 digits in length.

Up to 30 factors can be designated and are assigned a numeric code between 1 and 30 (by card location). If the TYPE 7 cards are read by the program, both TYPE 7A cards must also be read, even if they are both blank.

The required card format is --

Col 1, 2	'PM'
Col 3	'1' or '2'
Col 4-5	not used
Col 6-10	Constant 1 or 16
Col 11-15	Constant 2 or 17
⋮	⋮
Col 76-80	Constant 15 or 30

Notes: 1 -- Constants will be assigned an identification number from 1 to 30 depending on their position on the cards.

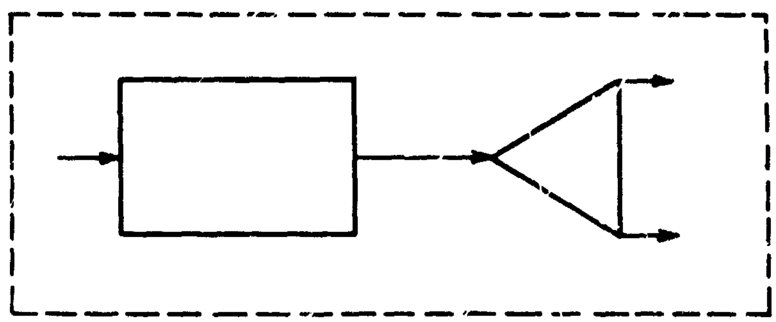
2 -- Both 'PM' cards must be present if any 'P' type cards are used.

8. Card Type No. 8 -- QUERY TYPE DESCRIPTION

These input cards, using a simple numeric "language," describe the anticipated path that a query will follow while being processed. Each query description requires three cards which may identify up to eighteen steps in the processing flow of a query. There are three different types of steps that can be utilized to describe the path of a query through the processing system. The first type is the NORMAL PROCESSING step which provides the basic method for expressing the flow of a particular query's processing. Although sequences of these "normal" steps may completely define many expected paths, two alternate steps have been provided that can be employed to still further expand the flexibility of the query description. These two alternates are the MULTIPLE DECISION and the SUBSTRING SELECTION steps. All three types of steps are detailed in the following subsections.

a. The Normal Processing Step

A normal step contains the elements of a flow diagram similar to those commonly employed by systems analysts or computer programmers, and can be illustrated as follows:



Such a step in a query's path is denoted within the structure provided by the following six items:

(1) the numeric code for the EVENT that is to process the query at that step;

(2) the numeric code of the TIME distribution to be employed in determining the amount of time the event is required to process the query;

(3) the numeric code of the appropriate MULTiplicative FACTOR;

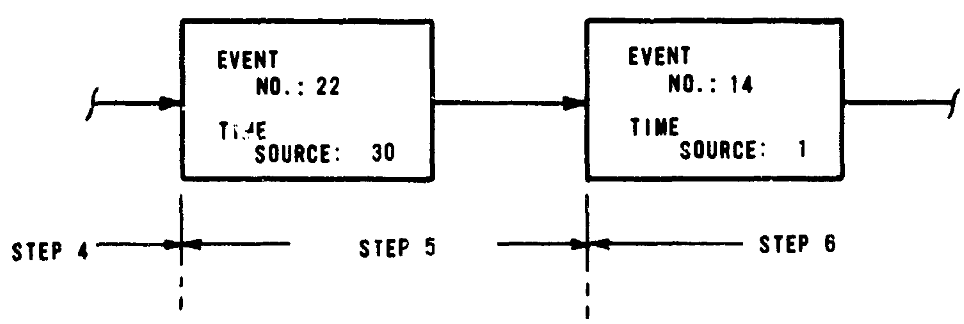
(4) the PROBability of the event FAILing to properly process the query;

(5) the next STEP if the event does FAIL to properly process the query; and

(6) the next STEP if the event SUCCESSfully processes the query.

The use of flow charts and simple forms can reduce the effort necessary to format this type of input data. One such form is illustrated in Figure 12.

The following block diagram illustrates a step in a query's processing which eliminates the decision block of the basic flow diagram since it is implied that both exits from the decision block go to the same place.



The interpretation of this charted flow is --

The fifth step in the processing of this query utilizes event number 22. The length of time required by event 22 to process the query type can be selected from time distribution number 30. When the query is completely processed at event 22, the next step in its processing path is STEP 6. The value of time selected from time distribution 30 is not to be multiplied by any factor. There is no probability of event 22 failing to process the query, and therefore there is no fail step.

This interpretation is indicated for punching onto the proper descriptor input card for the particular query type being charted as --

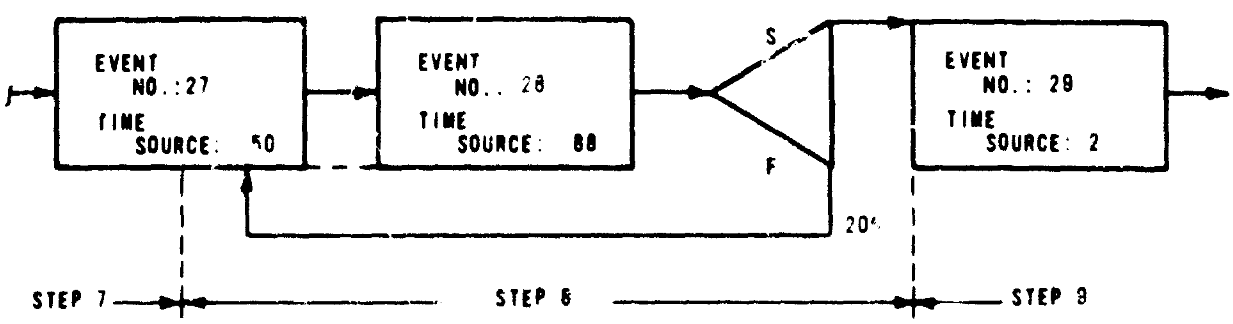
CARD 1 STEP	55	56	57	58	59	60	61	62	63	64	65	66	card column numbers
5	2	2	3	0								6	
	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP		

The event number can refer to any of the three types of events -- NORMAL, NQ, or INTERLOCKED. Since the successful step, STEP 6, is in normal sequential order, the number doesn't have to be entered. The program interprets all blank successful steps in the i^{th} step as meaning the $(i + 1)$ step. Had there been a need for some multiplicative factor, this would be indicated by some positive integer value between 1 and 30. This would be added simply as --

CARD 1 STEP	55	56	57	58	59	60	61	62	63	64	65	66	card column numbers
5	2	2	3	0		3							
	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP		

This does not mean to multiply the time value by 3, but rather multiply the time value by the number found in location 3. Thus, the multiplicative factor may be 3.0, or some other value such as 0.667. Notice that the number 6 was not entered into the success step this time.

The slightly more complex step which is represented by the basic flow diagram could be charted as --



The interpretation of this charted flow is --

The eighth step in the processing of this query utilizes event number 28. The length of time required by event 28 to process this query type can be selected from time distribution number 88. The time value is not multiplied by any factor. After the query has been completely processed through event 22, there is a 20% probability that this query type may not have been properly processed. If the query were properly processed, its next processing STEP is number 9; if the query were not properly processed, it returns to STEP 7 for reprocessing.

This can be expressed for punching as --

CARD 2 STEP	19	20	21	22	23	24	25	26	27	28	29	30
8	2	8	8	8			2	0		7		
	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP	

Again the SUCCESS STEP can be left blank. This same rule also applies to the FAIL STEP. Therefore, if the FAIL STEP is left blank in an ith step when there is a PROBability of FAILure figure, the program assumes the next step after failure is the (i + 1) step. Thus, leaving both the SUCCESS and FAIL STEPS blank with a PROBability of FAILure is the same as having no PROBability of FAILure.

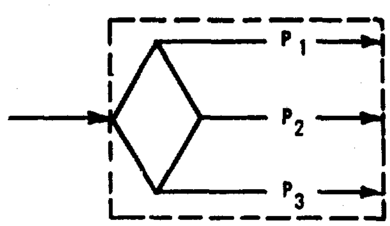
The FAIL and SUCCESS routes, at any step, may be designated as any integer between 1 and 18. Thus at the ith step, the FAIL and SUCCESS STEPS may indicate return to step i, (i-1), (i-2), etc., as well as advancing to step (i + 1), (i + 2), etc. These steps must, however, be within the interval of $1 \leq$ step # ≤ 18. Obviously the "PROB FAIL" need not be used to indicate only failure, rather it can be utilized to indicate any two-way decision point.

The numeric code 99 appearing in either a FAIL STEP or SUCCESS STEP indicates the termination of the query's processing. Several such codes may appear in any of the steps, but at least one terminating code mustappear somewhere in each query type decriptor or the program will not start. An example of the use of the code could be --

CARD 3 STEP												
	31	32	33	34	35	36	37	38	39	40	41	42
15	2	7	1	2			1	0	9	9	1	7
	43	44	45	46	47	48	49	50	51	52	53	54
16	3	3	2	5			2	0	1	8	9	9
	55	56	57	58	59	60	61	62	63	64	65	66
17	2	8	1	3							9	9
	67	68	69	70	71	72	73	74	75	76	77	78
18	3	4	2	5	2	1					9	9
	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP	

b. <u>The Multiple Decision Step</u>

The first alternate step provides the capability for a probabilistic selection among several SUCCESS or FAIL STEPS. These special steps are designated multiple decision blocks. A block diagram representation of this type of step can be illustrated as --



The fact that a multiple decision block is to be utilized and the location of the proper decision block are both denoted by the entry of a negative integer (between 1 and 9 inclusive) in the appropriate SUCCESS or FAIL STEP. The negative sign instructs the program to count from the bottom up rather than from the top down. Thus, -1 denotes step 18, -2 denotes step 17, etc. For this reason decision blocks can only be designated in steps 10 through 18 inclusive.

The elements of a multiple decision block which replace the normal step elements consist of --

(1) a possible acceptable processing step, "a";

(2) the probability of employing "a";

(3) a second possible acceptable processing step "b";

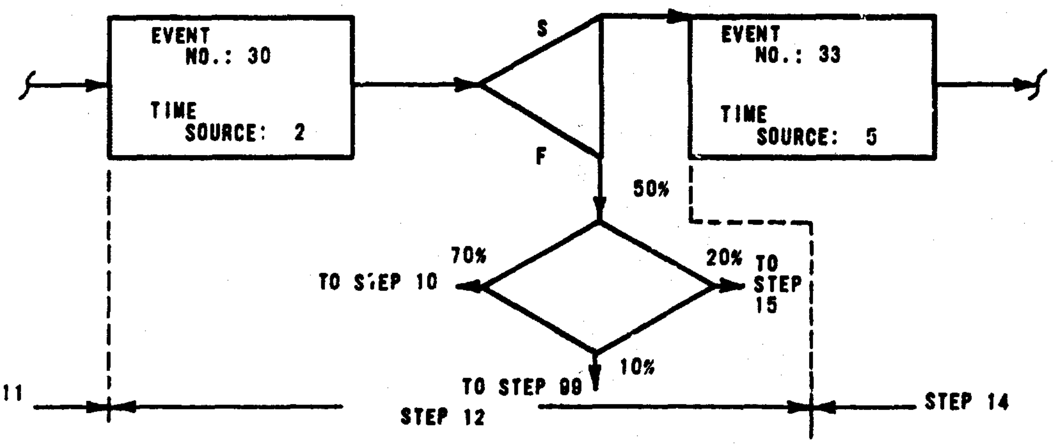
(4) the probability of employing "b";

(5) a third possible acceptable processing step "c"; and

(6) the probability of employing "c".

The acceptable processing steps need not be placed in any particular numeric order, and the block may contain less than three steps. However, the probabilities must sum to 100%. The first step denoting the need for a multiple decision block should numerically precede the step containing the block.

A flow chart of the application of a multiple decision step would be --



The interpretation of this charted flow is --

The twelfth step in the processing of this query utilizes event number 30. The length of time required by event 30 to process this query type can be selected from time distribution number 2. The time value is not multiplied by any factor. After the query has been completely processed through event 30, there is a 50% probability that this query type may not have been properly processed. If the query were properly processed, its next processing step is number 14; if the query were not properly processed, there is a 10% probability that it will be terminated, a 20% probability that it will go to step 15, and a 70% probability that it will go to step 10.

This can be expressed for punching as --

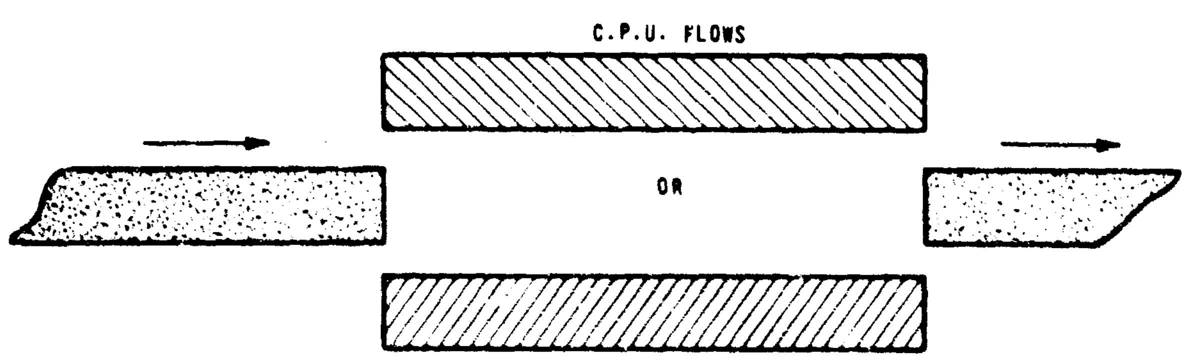
CARD 2 STEP	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP	
	67	68	69	70	71	72	73	74	75	76	77	78
12 (-7)	3	0		2			5	0	-	6	1	4

CARD 3 STEP	PROB		STEP		PROB		STEP		PROB		STEP	
	7	8	9	10	11	12	13	14	15	16	17	18
13 (-6)	7	0	1	0	1	0	9	9	2	0	1	5
	19	20	21	22	23	24	25	26	27	28	29	30
14 (-5)	3	3		5								
	EVENT NO.		TIME SOURCE		MULT FACT		PROB FAIL		FAIL STEP		SUCCESS STEP	

Of course, a multiple decision block can also be called by a SUCCESS STEP, as well as by one of the steps within a multiple decision block.

c. The Substring Selection Steps

Part of the basic rationale contended that a specific portion of a query's anticipated processing path may be independent of the abutting anterior or posterior portions as illustrated --



QUERY DESCRIPTION CARDS

QUERY IDENT | | | | |
GROUP TYPE

CARD 1

1	2	3	4	5	6
Q					1

	EVENT		TIME		MULT		PROB		FAIL STEP		SUCS STEP	
	7	8	9	10	11	12	13	14	15	16	17	18
STEP 1												
	19	20	21	22	23	24	25	26	27	28	29	30
2												
	31	32	33	34	35	36	37	38	39	40	41	42
3												
	43	44	45	46	47	48	49	50	51	52	53	54
4												
	55	56	57	58	59	60	61	62	63	64	65	66
5												
	67	68	69	70	71	72	73	74	75	76	77	78
6												

CARD 3

1	2	3	4	5	6
Q					3

STEP	EVENT		TIME		MULT		PROB		FAIL STEP		SUCS STEP	
	7	8	9	10	11	12	13	14	15	16	17	18
13 (-6)												
	19	20	21	22	23	24	25	26	27	28	29	30
14 (-5)												
	31	32	33	34	35	36	37	38	39	40	41	42
15 (-4)												
	43	44	45	46	47	48	49	50	51	52	53	54
16 (-3)												
	55	56	57	58	59	60	61	62	63	64	65	66
17 (-2)												
	67	68	69	70	71	72	73	74	75	76	77	78
18 (-1)												
ALT 1	STEP		PROB		STEP		PROB		STEP		PROB	
ALT 2	Q	Q	PROB		Q	Q	PROB		Q	Q	PROB	

CARD 2

1	2	3	4	5	6
Q					2

	EVENT		TIME		MULT		PROB		FAIL SUCS		SUCS STEP	
	7	8	9	10	11	12	13	14	15	16	17	18
STEP 7												
	19	20	21	22	23	24	25	26	27	28	29	30
8												
	31	32	33	34	35	36	37	38	39	40	41	42
9												
	43	44	45	46	47	48	49	50	51	52	53	54
10 (-9)												
	55	56	57	58	59	60	61	62	63	64	65	66
11 (-8)												
	67	68	69	70	71	72	73	74	75	76	77	78
12 (-7)												
ALT 1	STEP		PROB		STEP		PROB		STEP		PROB	
ALT 2	Q	Q	PROB		Q	Q	PROB		Q	Q	PROB	

NORMAL ENTRIES

EVENT: EVENT ID NUMBER
TIME: OGIVE ID NUMBER
MULT: MULTIPLICATIVE FACTOR
PROB: PERCENTAGE FAIL RATE
FAIL: NEXT STEP IF FAIL OCCURS
SUCS: NEXT STEP ON SUCCESS

EVENT 99 IS "COMPLETION"
EVENT 98 CALLS SUBDESCRIPTION
(NEXT 3 ENTRIES REFER TO ALT 2 TYPE STEPS)

USE NEGATIVE STEP NUMBERS FOR ALT 1 --
MULTIPLE PATH STEPS

ALTERNATE 1 ENTRIES

STEP: STEP NUMBER FOR FOLLOWING PROBABILITY
PROB: PERCENT PROBABILITY OF PATH FOLLOWING PREECEDING STEP

ALTERNATE 2 ENTRIES

QQ: QUERY SUBDESCRIPTION ID NUMBER
PROB: PERCENT PROBABILITY OF PATH FOLLOWING PRECEDING SUBDESCRIPTION

FIG. 12 QUERY DESCRIPTION FORM

The second alternate provides the capability for the probabilistic selection of a section from up to nine substrings for inclusion in the anticipated processing path of a query. This alternate utilizes several special steps, the first of which is the calling routine that monitors the selection and transfer of control to and from the selected substring. Such a special calling step is designated and initiated by the numeric code 98 appearing as the event number in any step. This special type of step then contains the following elements in place of the normal elements in a step --

(1) the numeric code 98;

(2) the number of the step containing from one to three numeric codes designating processing subpaths with their associated probabilities of selection;

(3) the number of the step containing from one to three additional numeric codes designating subpaths with their associated probabilities of selection;

(4) the number of the step containing from one to three additional numeric codes designating subpaths with their associated probabilities;

(5) the next step if a failure occurs in the selected subpath; and

(6) the next step if the processing in the selected subpath is successful.

The number of additional steps required for substring selection can vary from 1 to 3 as a function of the number of substrings to be employed. These steps, whose locations are designated as positive integer step numbers by the elements (2), (3), and (4) in the calling step, are variations of the multiple decision step. Instead of designating a possible acceptable processing step as an element, these decision steps designate the numeric code for a particular substring. Therefore, the elements contained in each of these types of decision steps consist of --

(1) a numeric code for some substring "a";

(2) the probability of selecting "a";

(3) a numeric code for some substring "b";

(4) the probability of selecting "b";

(5) a numeric code for some substring "c", and

(6) the probability of selecting "c".

If 1 to 3 substrings are to be considered, only one decision step is necessary; if 1 to 6 substrings are to be considered, two decision steps are necessary; and if 1 to 9 substrings are to be considered, then three decision steps are necessary. Thus, 5 substrings would require two decision steps while 7 substrings would require three. The step numbers of these decision steps must be numerically greater than the number of the calling step that references these steps. Therefore, if step 11 is the calling step, the numbers of any referenced decision steps must be 12 $\leq$ step # $\leq$ 18. The selection probabilities of all the referenced decision steps must total 100%.

One interpretation of a utilization of the substring selection steps could be --

The seventh step in the processing of the query requires the selection of one of nine subdescriptors which are numerically coded as 15, 25, 35, 45, 55, 65, 75, 85, and 95. The probability of selecting one of the subdescriptors is uniform. If there is an error within the subdescriptor, the query processing terminates. The next processing step is the next numerically available step which employs event number 33 and time source 4.

This alternate two entry can be expressed for punching as --

CARD 2 STEP	7	8	9	10	11	12	13	14	15	16	17	18
7	9	8		8		9	1	0	9	9	1	1
	19	20	21	22	23	24	25	26	27	28	29	30
8	2	5	1	1	1	5	1	1	4	5	1	1
	31	32	33	34	35	36	37	38	39	40	41	42
9	5	5	1	1	6	5	1	1	3	5	1	1
	43	44	45	46	47	48	49	50	51	52	53	54
10 (-9)	9	5	1	2	6	5	1	1	7	5	1	1
	55	56	57	58	59	60	61	62	63	64	65	66
11 (-8)	3	3		4								

Basic processing manipulations have been expressed in this numeric language for the simulation of a particular system such as the one diagrammed in Figure 12. However, a systems engineer applying his own ingenuity and often working with the model should be able to explain more complex and involved configurations.

The required card format for a normal step QUERY TYPE DESCRIPTION input is --

Col 1	'Q'	
Col 2-5	Query Type Identification Number	Col 2-3 GROUP Col 4-5 SUBTYPE
Col 6	Card number (1, 2, or 3)	
Col 7-8	Event for step 1, 7, or 13 (according to card #)	
Col 9-10	Time Source (Histogram # or blank for equation)	
Col 11-12	Multiplicative Factor (blank if factor is one)	
Col 13-14	Percent Failure Rate (may be blank for no failure)	
Col 15-16	Next step for failure (blank for next step)	
Col 17-18	Next step for success (blank for next step)	

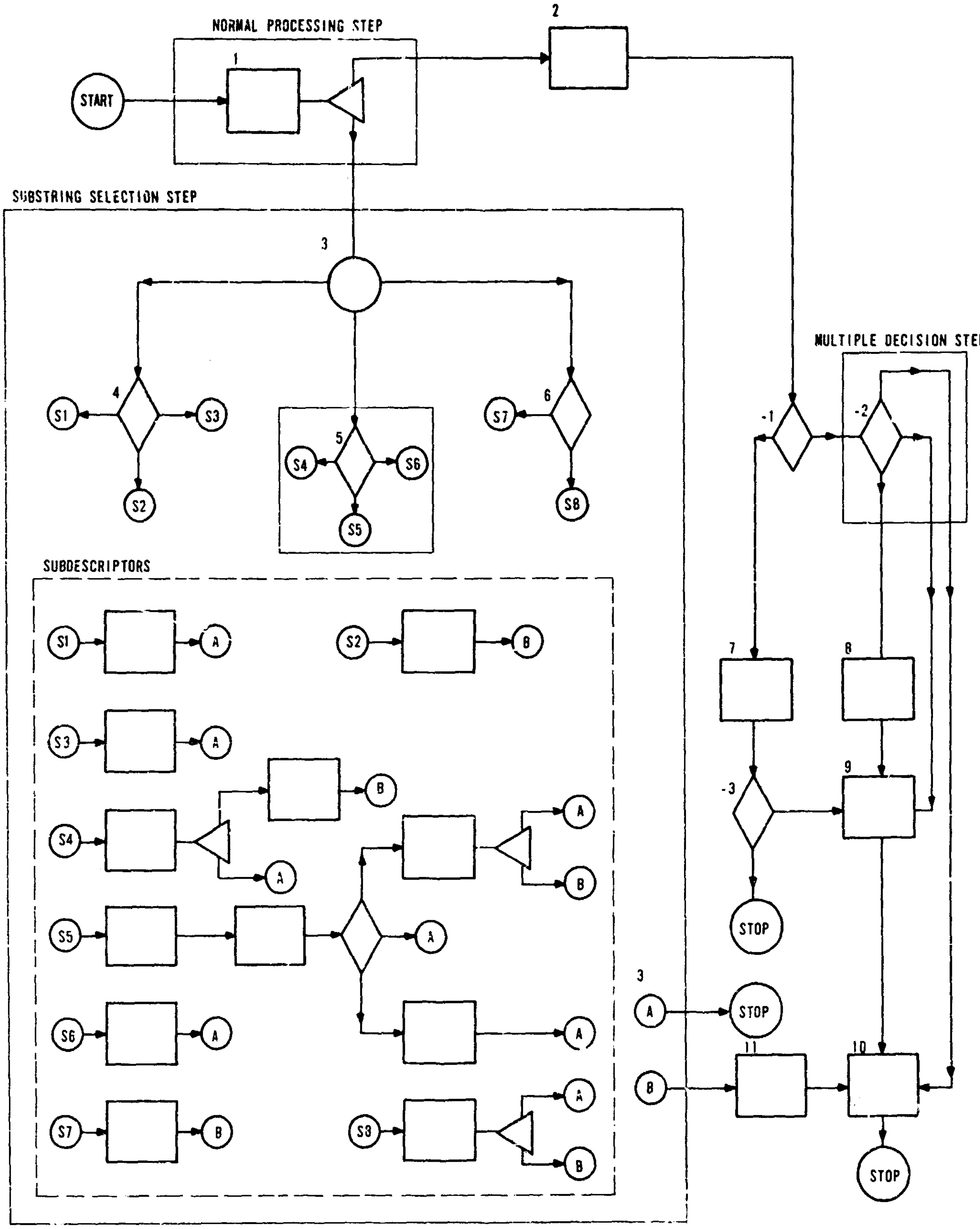


FIG. 13 EXAMPLE OF LOGIC FLOW THAT CAN BE DEPICTED WITHIN SIMULATION LANGUAGE

Card Type No. 8A -- QUERY SUBDESCRIPTIONS

These input cards, employing practically the same language used for the Type 8 cards, describe the anticipated subpaths a query may follow.[1] Up to a maximum of nine subdescriptions may be entered. Each query subdescription requires one card containing up to six descriptive flow steps for processing the query within the substring.

Only NORMAL PROCESSING and MULTIPLE DECISION[2] type steps can be utilized in the subdescription; SUBSTRING SELECTION steps are not allowed. Upon completion of the designated processing within the subdescriptor, the path assembling control is returned to the initiating calling step of the main descriptor. It should be noted that the termination of a query's processing path is only possible in the main descriptor.

Since SUBSTRING SELECTION steps and termination codes are not allowed in the subdescriptor, the numeric codes 98 and 99 have been used to indicate a return from the subdescriptor to the FAIL STEP or SUCCESS STEP respectively within the initiating main descriptor calling step. These return codes are placed in the desired FAIL STEP or SUCCESS STEP locations within any of the subdescription steps. The program checks for at least one 99 code in a subdescription before it will start. The interplay between the subdescriptor and main descriptor is given in the pictorial summary on the following page.

Event numeric codes in a subdescriptor can refer to the same type of events as well as the exact same events as those listed in the main descriptor. This is also true for the time distribution codes. However, all step numbers in the subpath must refer only to steps in the subdescriptor; no step number in a subdescriptor can refer to a specific step number in the main descriptor.

---

[1] If any Type 8 cards are read into the program, a Type 8A card must also be read.

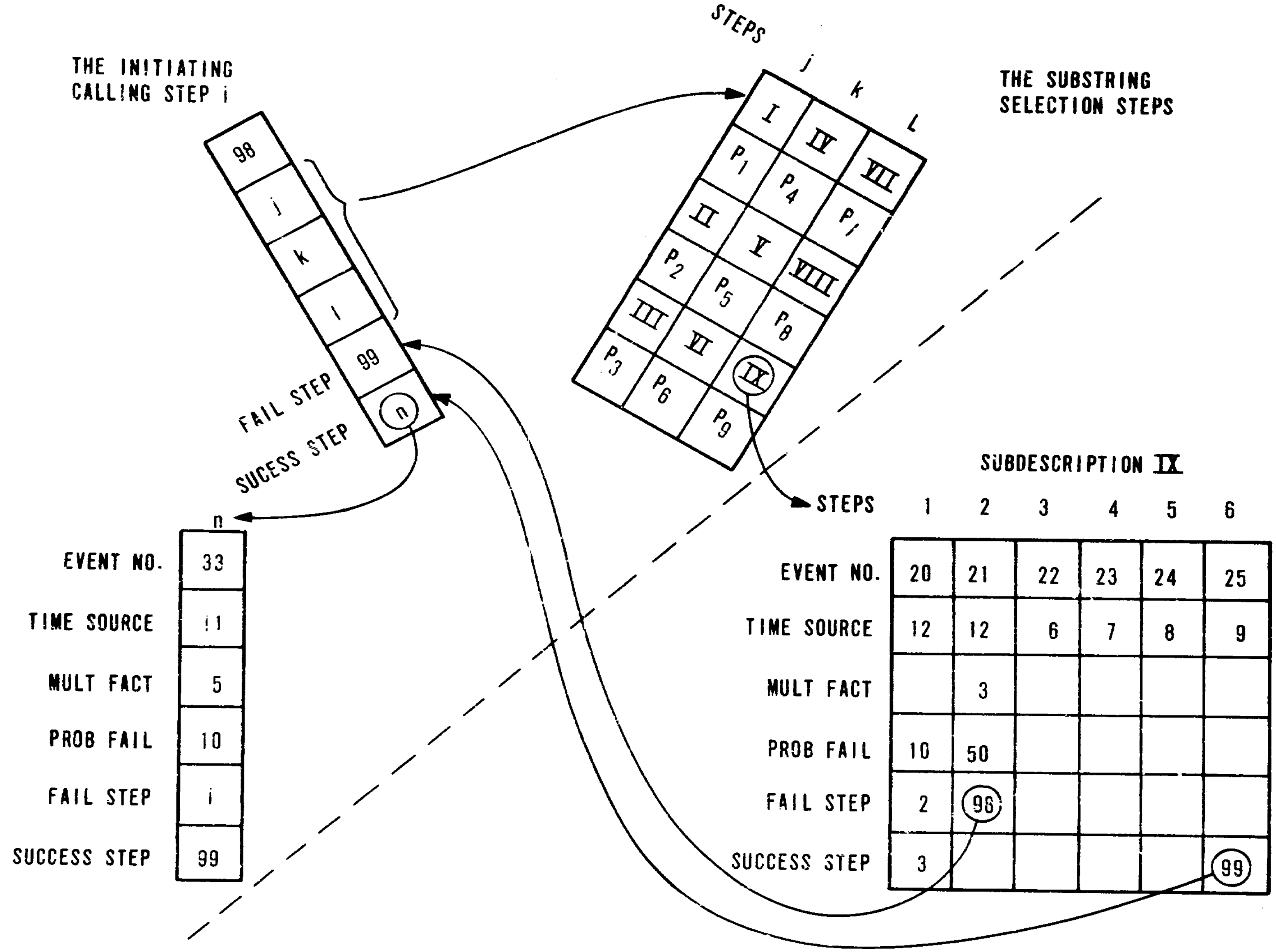
[2] A subdescriptor is restricted to a maximum of two MULTIPLE DECISION type steps since there are only a total of six descriptive flow steps.

Col 19-30	Repeat Col 7-18 for step 2, 8, or 14
Col 31-42	Repeat Col 7-18 for step 3, 9, or 15
Col 43-54	Repeat Col 7-18 for step 4, 10, or 16
Col 55-66	Repeat Col 7-18 for step 5, 11, or 17
Col 67-78	Repeat Col 7-18 for step 6, 12, or 18
Col 79-80	not used

Notes: 1 -- At least one success for fail step must be step 99 which signals end of processing for that query type.

2 -- Last Query Description Card must contain Q9999 in Cols 1-5 to signal end of Type 8 cards if fewer than 200 types are to be read.

3 -- A maximum of 200 query types is allowed (600 cards).



NOTE: THE RETURN CODES MAY BE PLACED IN EITHER THE SUBDESCRIPTOR FAIL STEP OF SUCCESS STEP

The required card format is

Col 1-2	'QQ'
Col 3-4	Query subdescription identification number
Col 5-6	not used
Col 7-78	Same as query description cards (Type 8)

Notes:

1 -- The last 'QQ' card must contain QQ99 in Cols 1-4 to signal end of Type 8A cards if fewer than 9 are to be read.

2 -- A maximum of 9 'QQ' cards may be used.

3 -- At least one success (or fail) step must be step <u>99</u> which signals return to main query description.

9. Card Type No. 9 -- ARRIVAL OF QUERIES

These input cards allow the systems engineer to express the query loading factor against the system. This factor represents the anticipated initiation of a quantity of queries against the system during some time interval. The length of a time interval is optional and may be expressed in positive integer values of specific Δt's. The distribution of query types over the intervals may be expressed with a uniform distribution, a normal distribution, or as a constant value. Up to 7 query types can be designated by a single card, and up to 400 cards can be read into the simulating program. However, 1000 queries are the maximum that the simulating program can handle per iteration. A quick program check is made by summing up the high range values for all the uniform distributions and the means and standard deviations for all the normal distributions to be sure this total is less than 1000. Another condition of the input is that the designated time intervals (identified by ranges of Δt) must be in ascending order.

The required card format is --

Col 1	'R'
Col 2-4	First time interval number (T_F)
Col 5-7	Last time interval number (T_L)
Col 8	not used
Col 9	"U" for uniform distribution; "N" for normal (Gaussian) distribution
Col 10	not used
Col 11-14	Query type number
Col 15-17	Minimum (or mean) number of this query type to be generated during $T_F \leq T \leq T_L$
Col 18-20	Maximum number (or standard deviation) of this query type to be generated during $T_F \leq T \leq T_L$

Repeat Cols 11-20 6 more times per card
(a total of 7 query types per card).

Notes: 1 -- Last Type 9 card must have R999 in Cols 1-4 if fewer than 400 cards are to be read.

2 -- A maximum of 400 'R' cards is allowed.

3 -- T_L must be greater than or equal to T_F on the same card and T_F on each card must be greater than or equal to T_F on the preceding card.

4 -- A maximum of 1000 queries per iteration is allowed. This is computed as $\Sigma U_{MAX} + \Sigma N_M + \Sigma N_S$

where U_{MAX} is the maximum number from a uniform distribution,

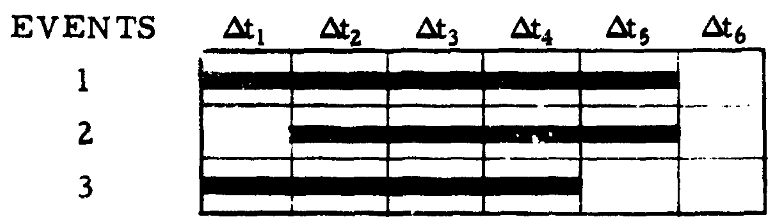
N_M is the mean for a normal distribution, and

N_S is the stand deviation for a normal distribution.

5 -- If a fixed number of a certain query type is to be generated, that number may be entered in Cols 15-17 and Cols 18-20 left blank.

10. Card Type No. 10 -- SCHEDULE OF OPERATION

These input cards enable the systems engineer to specify the work schedule of both the user and the system components available over the simulation time line. This is accomplished by identifying which events are scheduled to be available during each Δt interval. If two or more Δt intervals span a homogeneous event state, then a range of Δt's can be specified; e.g.,



INPUT DATA

schedule	events
1-1	1, 3
2-4	1, 2, 3
5-5	1, 2

If events are interlocked, only the central event, i.e., the event to which all the other events are interlocked, need be designated to schedule all these events. Up to 400 cards can be prepared (at most, this provides 1 card for every input Δt interval).

The required card format is --

Col 1	'S'
Col 2	not used
Col 3-5	First time interval number T_F
Col 6	not used
Col 7-9	Last time interval number T_L
Col 10	not used
Col 11-12	Event available during $T_F \leq T \leq T_L$

Col 13-14	Event available during $T_F \leq T \leq T_L$
⋮	⋮
Col 79-80	Event available during $T_F \leq T \leq T_L$.

Notes:

1 -- Last 'S' card must contain 999 in Col 3-5 to signal end of such cards if fewer than 400 are to be read.

2 -- A maximum of 400 'S' cards is allowed.

3 -- T_L must be greater than or equal to T_F on the same card and T_F, on each card must be greater than T_L on the preceding card; i. e., only one card may refer to a particular time interval.

11. Card Type No. 11 -- RUN LENGTH

This input card allows the systems engineer to specify some finite number of iterations to be performed. The card must always be read into the program when beginning the simulation from a cold start.

There is no need to read in a new RUN LENGTH card for a restart. However, if at any time during the simulation it is desired to alter the number of iterations to be performed, a new RUN LENGTH card can be read into the program by placing SENSE SWITCH 3 in the ON position after entering the card in the card reader.

The required card format is --

Col 1	'Z'
Col 2-5	The number of iterations desired
Col 6-80	not used

C. QUERY GENERATOR

The subroutines composing the QUERY GENERATOR begin the simulating procedure by initiating the number of each query type to be processed. Using a normal or uniform distribution or a fixed value designated by the QUERY ARRIVAL DISTRIBUTION cards, query types are appropriately generated for each specified time interval. The initiated time of each generated query type is determined by randomly selecting a time value within the defined time interval. This process is continually repeated until either all the defined time intervals have been processed or 1,000 queries have been generated for the iteration.

The queries are then sorted by initiating time (lowest value first) and then the information developed by the subroutines is printed. This information includes --

1. the number of each query type generated per time interval for the iteration and overall completed iterations; and
2. the number of queries of each type and the total number of queries generated during the iteration and overall completed iterations.

D. EVENT SEQUENCE GENERATOR

These subroutines combine to determine the events, operating times, and flow sequences that produce a processing path for each generated query. The determinations are governed by the information contained on the QUERY DESCRIPTIONS and SUBDESCRIPTIONS, the LIST OF EVENTS, and the PROBABILITY DISTRIBUTIONS and MULTIPLICATIVE FACTORS input cards.

The basic algorithm for interpreting every descriptor step to assemble the processing path of each query initiated during an iteration is as follows:

1. identify the event required for the processing of the query;
2. determine the amount of time which the event requires to process the query; and
3. denote the proper sequence for designating the next processing event.

Each event and its required operating time are collectively defined as a stage in the processing path of a query. A completely assembled processing path for any query, then, is a string of stages. The termination of an assemblage is caused by either the logical end of processing denoted by the code 99 in the QUERY DESCRIPTION, or when the number of stages assembled in a path totals 60. In this latter case, a message is printed indicating which query's processing path has exceeded the 60 stages allowed. The simulation will continue even though some quer paths may be incomplete.

During the assembling of the stages, all interlocked events are assigned to a stage under the central event number. Events having the same event number in abutting stages are collapsed into one stage and their operating times are combined. If the sum of the operating times exceeds a 4-digit value, a warning message is printed indicating the event. The figure 9999 is then substituted as the required operating time.

After all the processing paths have been completely assembled, pertinent data generated by these subroutines are printed. This information includes:

1. the total number of events utilized and their associated processing time for all the queries generated;
2. a listing of all the types of events with their total work time for the iteration and overall completed iterations; and
3. the percentage of the work time contributed by all interlocked events to the central event for the iteration and overall completed iterations.

E. SEQUENCE INTEGRATOR

This subroutine performs the role of a scheduling director, analyzing the data flow in the retrieval process and assigning work units to available equipment and personnel.

The basic algorithm employed involves --

1. selecting the query processing path with the earliest availability time;
2. assigning the first available service unit of the event required in the processing of the query for the indicated period of time; and
3. determining the amount of delay time encountered by the query before it is processed.

Obviously the time-wise integration of the processing stages of a set of queries is, in reality, more complex than an algorithm may indicate. Involved in the integration are interations of the various processing paths and the availability of processing events as dictated by the SCHEDULE OF OPERATIONS input cards and as altered by down time due to maintenance. Any query processing integrant may be placed in queue because a required event is not available. The event may not be available because it is not scheduled, it is down for repair or maintenance, or it is processing other data. The program delays each integrant in queue while the service units for the necessary operation are unavailable. Each integrant is assigned out of queue on a first-come-first-served basis to the first available service unit in a string.

The size of the memory core of the computer used to process the simulation program imposes some constraints on the amount of data that can be simultaneously processed. Consequently, up to a maximum of 50 query paths are considered by the integrator program at any one time[1]. However, it may be possible that a

[1] Path segments (a portion of the query processing path containing 12 or less stages) of the first 50 queries to be generated are loaded from their disk storage sectors into core for process integration. The reader is reminded that a stage is composed of an event number and the required operating time, and that a query processing path is an ordered sequence of up to 60 stages. As all of the stages in a segment are completely integrated, the next sequential segment is transferred into core in its place. Whenever a query's path has been completely integrated (this may take place in any of the five path segments), the first segment of a path of a new query is loaded into core. This process continues until the paths of all generated queries have been transferred into core or the integration over the time line is completed.

51st query path (or a whole series of additional query paths) should also be integrated. In this instance, the program will unrealistically accumulate delay time against the 51st (and successive) query. If, however, the systems engineer identifies and schedules the user initiating the query as the first event in the QUERY DESCRIPTION, the unrealistic delay time can be readily identified as delay preceding this event.

F. PROGRAM RESTART OR REITERATE

This subroutine houses the iterative processing control. The simulating program continues to operate until a designated number of iterations have been completed, thereby allowing computations over periods of time without constant operator monitoring.

By utilizing selected sense switches, the operator of the simulating program can make alterations to the program's iterative processing control after the simulation has been initiated without aborting the effort. SENSE SWITCH 3, when in the ON position, will cause the simulating program to read a new RUN LENGTH card which could increase or decrease the total number of desired iterations. SENSE SWITCH 4, when in the ON position, initiates a typed message -- "GISMO ITERATION · · · DONE -- PRESS START TO CONTINUE" -- and then the C. P. U. pauses until the operator either continues the iterative processing by pushing the START BUTTON, or terminates the simulation.

Because of the structure of the simulating program, it is possible to include a unique restarting capability. If for any reason it is necessary to interrupt the simulation at the end of an iteration before the simulation has been completed, the simulating program can be restarted at the same point after some length of elapsed time without aborting the effort. This is possible since the management of all the program's data is performed and contained separately by the second disc, thereby divorced of the operation of the first disc which contains the monitoring routines and the simulating program instructions. When restarting, the INPUT segments are simply bypassed by calling for the subroutine GISGO instead of GISMO, which returns the simulation to the beginning point for the start of the next iteration. Thus to the simulating program, no time interval has elapsed.

APPENDIX B

EXAMPLES OF THE SIMULATION I/O DISPLAYS

(The actual outputs have been slightly altered in order to provide a more efficient printing of the material.)

INPUT DATA SPECIFICATIONS

	CARD TYPES	COLS 1, 2	NUMBER OF CARDS
1 -	Identification	ID	1 Card
2 -	Control	K	1 Card
3 -	List of events	L	2-35 Cards (1 per event)
4 -	Meaning of query types	M	2-200 Cards (1 per Q. T.)
5 -	System input definition	N	1 Card
6 -	Summary output definition	O	2 Cards
7 -	Probability distributions	P	3-98 Cards (2 per ogive)
7A -	Multiplicative constants	PM	2 Cards
8 -	Query description	Q	4-600 Cards (3 per query)
8A -	Query subdescriptions	QQ	1-9 Cards (1 per subsection)
9 -	Query arrival distribution	R	2-400 Cards
10 -	Event operation schedule	S	2-400 (1 per input time interval)
11 -	Run length	Z	1 Card

INPUT LIMITATIONS

Events:	35 maximum
Service Units:	100 max -- any distribution per event
Query Types:	200 maximum
Ogives:	49 maximum
Steps:	18 max per Query type (exclusive of subsections) 6 max per Query subsection
Total # of Queries processed per iteration:	1000 maximum
Input Times Sectors:	400 maximum
Stages per Query:	60 maximum
Output Time Sectors:	25 maximum
Query Subsections:	9 maximum
Multiplicative Constants:	30 maximum
Decision Blocks:	9 maximum per Query type (exclusive of subsections) 2 maximum per Query subsection
Query Groupings:	99 Query types maximum per group; maximum of 49 group.

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO

*** GISMO ***

A GENERAL INFORMATION SYSTEM MODEL

LIST OF EVENT CODES AND THEIR MEANING

EVENT CODE	MEANING		UNITS	MAINT. PROB	LOCKED TO
11	USER REQUESTS DATA BY PHONE		1	0	0
12	USER FORMULATES QUESTION FOR DISPATCH	NQ	1	0	0
13	USER FORMULATES QUERY FOR TELETYPE	NQ	1	0	0
14	USER RECEIVES OUTPUT DATA	NQ	1	0	0
21	COURIER DELIVERS REQUESTS	NQ	1	0	0
22	COURIER DELIVERS OUTPUT DATA TO USERS	NQ	1	0	0
30	SYSTEM CONSULTANT		1	1	0
31	CONSULTANT PREPARES QUERY		1	0	30
32	CONSULTANT CHECKS TELETYPE COPY		1	0	30
33	CONSULTANT CORRECTS HARD COPY		1	0	30
40	SATELLITE COMPUTER A		1	3	0
41	PAPER TAPE TO CARD CONVERSION--A		1	0	40
42	OFF-LINE PRINT--A		1	0	40
50	SATELLITE COMPUTER B		1	3	0
51	OFF-LINE PRINT--B		1	0	50
52	REMOTE PRINT--B		1	0	50
60	CENTRAL PROCESSOR		1	1	0
61	ENTER QUERY + SET-UP		1	0	60
62	FILE SEARCH		1	0	60
63	SORT/EDIT		1	0	60
64	SUMMARY		1	0	60
65	ON-LINE CORRECT		1	0	60
72	TELETYPE		2	2	0
74	PUNCH QUERY CARDS		1	1	0
76	VERIFY QUERY CARDS		1	1	0
81	TRANSPORT CARDS/TAPE FOR PROCESSING	NQ	1	0	0
82	TRANSPORT DATA FOR CORRECTION	NQ	1	0	0
83	TRANSPORT OUTPUT FOR DELIVERY	NQ	1	0	0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO INPUT

QUERY DESCRIPTION BY CODE NUMBER

QUERY CODE	MEANING SITE	METHOD OF INQUIRY	SEARCH	OUTPUT COMMUNICATIONS
0				
1101	A	TELEPHONE	LOW	COURIER/DATA LINK
1102	A	COURIER	LOW	COURIER
1201	A	TELEPHONE	HIGH	COURIER/DATA LINK
1202	A	COURIER	HIGH	COURIER
2201	B	TELEPHONE	HIGH	COURIER/DATA LINK
2202	B	COURIER	HIGH	COURIER
2303	B	TELETYPE	LO-HI	DATA LINK
3101	C	TELEPHONE	LOW	COURIER
3102	C	COURIER	LOW	COURIER
3303	C	TELETYPE	LO-HI	DATA LINK

A TOTAL OF 11 QUERY TYPES

6 QUERY GROUPS ARE USED - THESE ARE

11 12 22 23 31 33

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO　　INPUT

THE BASIC UNIT OF TIME IS MINUTES

THE NUMBER OF TIME SECTORS IN THE INPUT IS 200

THE DURATION OF EACH INPUT TIME SECTOR IS 60. MINUTES

THE SIMULATED TIME PERIOD THEREFORE IS 12000. MINUTES

THE OUTPUT IS DIVIDED INTO 6 INTERVALS

INTERVAL	FROM (SECTORS)	TO (SECTORS)	FROM (MINUTES)	TO (MINUTES)
1	8	31	420.	1860.
2	32	55	1860.	3300.
3	56	79	3300.	4740.
4	80	103	4740.	6180.
5	104	127	6180.	7620.
6	176	200	10500.	12000.

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO INPUT

OGIVE DESCRIPTIONS

OGIVE	PROB	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1		1	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	5
2		3	3	4	4	5	5	6	6	6	6	7	7	7	7	8	8	9	9	10	10
3		4	5	6	7	7	8	9	9	10	10	10	10	11	11	12	13	15	17	19	21
4		2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6
5		1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5
6		2	2	2	3	3	3	3	4	4	4	4	4	4	5	5	5	5	6	6	6
7		3	3	4	4	4	5	5	5	6	6	6	7	10	13	16	19	22	25	28	31
8		1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4
9		1	1	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	6	6
11		1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	3	3	4	5
12		2	4	6	7	8	9	10	11	13	15	16	17	18	20	25	30	35	40	45	50
13		40	42	44	46	48	50	52	54	56	58	60	60	65	70	75	80	85	90	95	100
14		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15		3	3	3	5	5	5	5	5	8	8	8	10	10	10	10	10	13	13	15	15
16		15	15	15	18	18	18	18	18	20	20	20	25	25	30	30	35	35	40	40	45
17		1	1	1	2	2	2	3	3	3	3	4	4	4	5	5	5	6	6	7	8
18		4	4	4	4	5	5	5	6	6	6	6	6	6	7	7	7	8	8	10	10
19		1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
20		1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5
21		1	1	1	1	2	2	2	3	3	4	4	4	4	4	5	5	6	6	8	10

A TOTAL OF 20 DIFFERENT OGIVES

THE FOLLOWING CONSTANTS ARE USED AS MULTIPLIERS

1	2	3	4	5	6	7	8	9	10
1.7	2.0	10.0	30.0	0.0	0.0	0.0	0.0	0.0	0.0
11	12	13	14	15	16	17	18	19	20
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	22	23	24	25	26	27	28	29	30
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO INPUT

QUERY TYPE DESCRIPTION

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1101	EVENT	11	31	74	76	81	98	1	82	42	51	52	83	22	14	9	0	0	0
	TIMING	1	6	5	5	9	7	10	9	12	12	12	9	4	14	20	0	0	0
	FACTOR	0	0	1	0	0	0	2	0	0	2	2	0	0	0	10	0	0	0
	FAIL PCT	0	0	0	6	0	0	30	0	0	0	0	0	0	0	30	0	0	0
	FAIL RTE	0	0	0	3	0	8	3	0	0	0	0	0	0	0	11	0	0	0
	SUCS RTE	0	0	0	0	0	-4	60	2	12	12	14	0	0	99	50	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1201	EVENT	11	31	74	76	81	98	4	82	42	51	52	83	22	14	9	0	0	0
	TIMING	1	6	5	5	9	7	20	9	13	13	13	9	4	14	50	0	0	0
	FACTOR	0	0	1	0	0	0	5	0	0	2	2	0	0	0	10	0	0	0
	FAIL PCT	0	0	0	6	0	0	20	0	0	0	0	0	0	0	30	0	0	0
	FAIL RTE	0	0	0	3	0	8	6	0	0	0	0	0	0	0	11	0	0	0
	SUCS RTE	0	0	0	0	0	-4	60	2	12	12	14	0	0	99	20	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1102	EVENT	12	21	31	74	76	81	98	1	82	42	51	83	22	14	10	0	0	0
	TIMING	2	4	7	5	5	9	8	10	9	12	12	9	4	14	70	0	0	0
	FACTOR	0	0	0	1	0	0	0	2	0	0	2	0	0	0	11	0	0	0
	FAIL PCT	0	0	0	0	6	0	0	20	0	0	0	0	0	0	30	0	0	0
	FAIL RTE	0	0	0	0	4	0	9	3	0	0	0	0	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	0	0	-4	70	3	12	0	0	0	99	0	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1202	EVENT	12	21	31	74	76	81	98	4	82	42	51	83	22	14	10	0	0	0
	TIMING	2	4	7	5	5	9	8	10	9	13	13	9	4	14	60	0	0	0
	FACTOR	0	0	0	1	0	0	0	5	0	0	2	0	0	0	11	0	0	0
	FAIL PCT	0	0	0	0	6	0	0	10	0	0	0	0	0	0	40	0	0	0
	FAIL RTE	0	0	0	0	4	0	9	6	0	0	0	0	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	0	0	-4	80	3	12	0	0	0	99	0	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2201	EVENT	11	31	74	76	81	98	4	82	42	51	52	83	22	14	9	0	0	0
	TIMING	1	6	5	5	9	7	15	9	13	13	13	9	4	14	20	0	0	0
	FACTOR	0	0	1	0	0	0	5	0	0	2	2	0	3	0	10	0	0	0
	FAIL PCT	0	0	0	6	0	0	25	0	0	0	0	0	0	0	20	0	0	0
	FAIL RTE	0	0	0	3	0	8	6	0	0	0	0	0	0	0	11	0	0	0
	SUCS RTE	0	0	0	0	0	-4	60	2	12	12	14	0	0	99	60	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2202	EVENT	12	21	31	74	76	81	98	4	82	42	51	83	22	14	10	0	0	0
	TIMING	2	4	7	5	5	9	8	15	9	13	13	9	4	14	50	0	0	0
	FACTOR	0	3	0	1	0	0	0	5	0	0	2	0	3	0	11	0	0	0
	FAIL PCT	0	0	0	0	6	0	0	15	0	0	0	0	0	0	50	0	0	0
	FAIL RTE	0	0	0	0	4	0	9	6	0	0	0	0	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	0	0	-4	70	3	12	0	0	0	99	0	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2303	EVENT	13	72	32	81	41	33	74	76	81	98	1	4	82	31	52	14	0	0
	TIMING	3	5	8	9	11	6	5	5	9	11	10	5	9	6	12	[illegible]	0	0
	FACTOR	0	0	0	0	0	0	1	0	0	12	2	5	0	0	2	0	0	0
	FAIL PCT	0	0	12	0	0	0	0	6	0	0	10	15	0	0	0	0	0	0
	FAIL RTE	0	0	6	0	0	0	0	7	0	13	3	6	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	10	0	0	0	0	15	40	20	0	7	0	99	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3101	EVENT	11	31	74	76	81	98	1	82	42	51	83	22	14	9	0	0	0	0
	TIMING	1	6	5	5	9	7	20	9	12	12	9	4	14	50	0	0	0	0
	FACTOR	0	0	1	0	0	0	2	0	0	2	0	4	0	10	0	0	0	0
	FAIL PCT	0	0	0	6	0	0	40	0	0	0	0	0	0	50	0	0	0	0
	FAIL RTE	0	0	0	3	0	8	3	0	0	0	0	0	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	0	-5	40	2	11	0	0	0	99	0	0	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3102	EVENT	12	21	31	74	76	81	98	1	82	42	51	83	22	14	10	0	0	0
	TIMING	2	4	7	5	5	9	8	10	9	12	12	9	4	14	50	0	0	0
	FACTOR	0	4	0	1	0	0	0	2	0	0	2	0	4	0	11	0	0	0
	FAIL PCT	0	0	0	0	6	0	0	30	0	0	0	0	0	0	50	0	0	0
	FAIL RTE	0	0	0	0	4	0	9	3	0	0	0	0	0	0	0	0	0	0
	SUCS RTE	0	0	0	0	0	0	-4	60	3	12	0	0	0	99	0	0	0	0

TYPE	STEP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3303	EVENT	13	72	32	81	41	33	11	31	74	76	81	98	1	4	82	52	14	7
	TIMING	3	5	8	9	11	6	1	6	5	5	9	13	20	20	9	12	14	10
	FACTOR	0	0	0	0	0	0	0	0	1	0	0	14	2	5	0	2	0	9
	FAIL PCT	0	0	12	0	0	0	0	0	0	6	0	0	15	15	0	0	0	90
	FAIL RTE	0	0	6	0	0	0	0	0	0	9	0	15	3	6	0	0	0	0
	SUCS RTE	0	0	0	0	12	-1	0	0	0	0	0	16	10	20	8	0	99	0

A TOTAL OF 10 DIFFERENT QUERY TYPES

QUERY SUBDESCRIPTIONS

TYPE	STEP	1	2	3	4	5	6
1	EVENT	61	62	65	0	0	0
	TIMING	8	15	21	0	0	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	15	40	0	0	0
	FAIL RTE	0	3	98	0	0	0
	SUCS RTE	0	99	99	0	0	0

TYPE	STEP	1	2	3	4	5	6
2	EVENT	61	62	63	65	0	0
	TIMING	8	15	17	21	0	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	0	15	40	0	0
	FAIL RTE	0	0	4	98	0	0
	SUCS RTE	0	0	99	99	0	0

TYPE	STEP	1	2	3	4	5	6
3	EVENT	61	62	63	64	65	0
	TIMING	8	15	17	19	21	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	0	0	15	40	0
	FAIL RTE	0	0	0	5	98	0
	SUCS RTE	0	0	0	99	99	0

TYPE	STEP	1	2	3	4	5	6
4	EVENT	61	62	65	0	0	0
	TIMING	8	16	21	0	0	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	15	40	0	0	0
	FAIL RTE	0	3	98	0	0	0
	SUCS RTE	0	99	99	0	0	0

TYPE	STEP	1	2	3	4	5	6
5	EVENT	61	62	63	65	0	0
	TIMING	8	16	18	21	0	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	0	15	40	0	0
	FAIL RTE	0	0	4	98	0	0
	SUCS RTE	0	0	99	99	0	0

TYPE	STEP	1	2	3	4	5	6
6	EVENT	61	62	63	64	65	0
	TIMING	8	16	18	20	21	0
	FACTOR	0	0	0	0	0	0
	FAIL PCT	0	0	0	15	40	0
	FAIL RTE	0	0	0	5	98	0
	SUCS RTE	0	0	0	99	99	0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO INPUT

EXPECTED ARRIVAL OF QUERIES ACCORDING TO TIME INTERVAL

(TIME INTERVALS ARE INCLUSIVE)

FROM	TO		Q	N	N	Q	N	N	Q	N	N	Q	N	N	Q	N	N	Q	N	N	Q	N	N
8	11	U	1101	0	1	1201	0	1	1102	0	1	1202	0	1	2201	0	1	2202	0	1	2303	0	2
8	11	U	3101	0	1	3102	0	1	3303	0	2	0	0	0	0	0	0	0	0	0	0	0	0
13	16	U	2303	0	3	3303	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	35	U	1101	0	1	1201	0	1	2201	0	1	2303	0	1	3101	0	1	3303	0	1	0	0	0
37	40	U	1101	0	1	1201	0	1	2201	0	1	2303	0	1	3101	0	1	3303	0	1	0	0	0
56	59	U	1101	0	1	1201	0	1	1102	0	1	1202	0	1	2201	0	1	2202	0	1	2303	0	1
56	59	U	3102	0	1	3303	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	64	U	1101	0	1	1201	0	1	1102	0	1	1202	0	1	2201	0	1	2202	0	1	2303	0	1
61	64	U	3102	0	1	3303	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	83	U	1101	0	2	1201	0	1	1102	0	1	2201	0	1	2303	0	2	3101	0	1	3303	0	1
85	88	U	1101	0	2	1201	0	1	1102	0	1	2201	0	1	2303	0	2	3101	0	1	3303	0	2
104	107	U	1101	0	1	2201	0	1	2303	0	2	3303	0	[illegible]	0	0	0	0	0	0	0	0	0
109	112	U	1101	0	1	2303	0	2	3303	0	2	0	0	0	0	0	0	0	0	0	0	0	0
176	179	U	1101	0	1	1201	0	1	1102	0	1	1202	0	1	2201	0	1	2202	0	1	2303	0	2
176	179	U	3101	0	1	3102	0	1	3303	0	2	0	0	0	0	0	0	0	0	0	0	0	0
181	184	U	2303	0	3	3303	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BETWEEN 0 AND 96 QUERIES ARE TO BE GENERATED FROM A UNIFORM DISTRIBUTION

AND AN AVERAGE OF 0 FROM A NORMAL DISTRIBUTION

WITH A STANDARD DEVIATION OF 0

EVENT AVAILABILITY ACCORDING TO TIME INTERVAL

(TIME INTERVALS ARE INCLUSIVE)

FROM	TO	EVENTS AVAILABLE																																		
1	2	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	5	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	11	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	16	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	17	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	20	30	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	22	81	82	83	74	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	24	81	82	83	74	76	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	26	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	27	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	29	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	30	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	31	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	35	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	40	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	41	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	44	30	81	82	83	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	46	81	82	83	74	76	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	48	81	82	83	74	76	60	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	50	50	60	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51	51	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	53	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	54	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	55	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	59	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	64	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	65	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	68	30	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	70	81	82	83	74	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
71	72	81	82	83	74	76	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	74	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	75	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	77	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	78	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	79	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	83	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	88	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	89	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	92	30	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
93	94	81	82	83	74	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	96	81	82	83	74	76	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
97	98	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
99	99	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	101	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102	102	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
103	103	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104	107	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109	112	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
113	113	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
114	116	30	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
117	118	81	82	83	74	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
119	120	81	82	83	74	76	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
169	170	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
171	171	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
172	173	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
174	174	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
175	175	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
176	179	72	11	12	13	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
181	184	72	11	12	13	14	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
185	185	30	21	40	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
186	188	30	81	82	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
189	190	81	82	83	74	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
191	192	81	82	83	74	76	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
193	194	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
195	195	40	50	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
196	197	40	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
198	198	81	82	83	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
199	199	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200	200	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PERCENT OF OUTPUT TIME SECTORS THAT EVENTS ARE AVAILABLE

FROM	TO	11	12	13	14	21	22	30	31	32	33	40	41	42	50	51	52	60	61	62	63	64	65	72	74	76	81	82	83
8	31	33	33	33	33	4	4	33	0	0	0	17	0	0	25	0	0	21	0	0	0	0	0	33	17	17	38	38	38
32	55	33	33	33	33	4	4	33	0	0	0	17	0	0	25	0	0	21	0	0	0	0	0	33	17	17	38	38	36
56	79	33	33	33	33	4	4	33	0	0	0	17	0	0	25	0	0	21	0	0	0	0	0	33	17	17	38	38	38
80	103	33	33	33	33	4	4	33	0	0	0	17	0	0	25	0	0	21	0	0	0	0	0	33	17	17	38	38	38
104	127	33	33	33	33	4	0	33	0	0	0	4	0	0	0	0	0	8	0	0	0	0	0	33	17	17	33	33	33
176	200	32	32	32	36	4	4	32	0	0	0	16	0	0	24	0	0	20	0	0	0	0	0	32	16	16	36	36	36

ACTUAL ARRIVAL OF QUERIES ACCORDING TO TIME INTERVAL

1ST LINE THIS ITERATION 2ND LINE SIMULATION TOTALS

(TIME INTERVALS ARE INCLUSIVE)

FROM	TO	Q	N	Q	N	Q	N	Q	N	Q	N	Q	N	Q	N	SUB TOTAL
8	11	1101	1	1201	0	1102	1	1202	1	2201	1	2202	0	2303	1	5
			1.		0.		2.		2.		2.		1.		2.	10.
8	11	3101	1	3102	1	3303	0	0	0	0	0	0	0	0	0	2
			2.		1.		1.		0.		0.		0.		0.	4.
3	16	2303	1	3303	2	0	0	0	0	0	0	0	0	0	0	3
			1.		3.		0.		0.		0.		0.		0.	4.
32	35	1101	0	1201	0	22[illegible]	1	2303	0	3101	1	3303	0	0	0	2
			1.		1.		1.		0.		1.		1.		0.	5.
37	40	1101	0	1201	0	2201	1	2303	0	3101	1	3303	1	0	0	3
			0.		1.		1.		1.		1.		2.		0.	6.
56	59	1101	0	1201	0	1102	0	1202	0	2201	1	2202	0	2303	1	2
			1.		0.		1.		0.		1.		0.		1.	4.
56	59	3102	1	3303	1	0	0	0	0	0	0	0	0	0	0	2
			2.		2.		0.		0.		0.		0.		0.	4.
61	64	1101	0	1201	0	1102	0	1202	1	2201	1	2202	1	2303	1	4
			1.		1.		0.		2.		2.		2.		2.	10.
61	64	3102	0	3303	0	0	0	0	0	0	0	0	0	0	0	0
			1.		0.		0.		0.		0.		0.		0.	1.
80	83	1101	1	1201	1	1102	1	2201	1	2303	2	3101	0	3303	1	7
			1.		1.		2.		1.		4.		0.		2.	11.
85	88	1101	1	1201	1	1102	0	2201	0	2303	0	3101	0	3303	1	3
			1.		2.		0.		0.		1.		0.		3.	7.
104	107	1101	0	2201	1	2303	1	3303	0	0	0	0	0	0	0	2
			0.		1.		2.		1.		0.		0.		0.	4.
109	112	1101	0	2303	1	3303	0	0	0	0	0	0	0	0	0	1
			0.		3.		1.		0.		0.		0.		0.	4.
176	179	1101	0	1201	0	1102	0	1202	1	2201	0	2202	1	2303	1	3
			0.		0.		1.		1.		0.		2.		2.	6.
176	179	3101	0	3102	1	3303	1	0	0	0	0	0	0	0	0	2
			0.		1.		2.		0.		0.		0.		0.	3.
181	184	2303	2	3303	3	0	0	0	0	0	0	0	0	0	0	5
			2.		6.		0.		0.		0.		0.		0.	8.

QUERIES GENERATED IN THIS ITERATION AND TOTALS

TYPE	NUM	TOTAL	TYPE	NUM	TOTAL	TYPE	NUM	TOTAL	TYPE	NUM	TOTAL	TYPE	NUM	TOTAL
1101	3	6.	1201	2	6.	1102	2	6.	1202	3	5.	2201	7	9.
2202	2	5.	2303	11	21.	3101	3	4.	3102	3	5.	3303	10	24.

TOTAL FOR THIS ITERATION 46.

TOTAL FOR THE SIMULATION 91.

EVENT SEQUENCE GENERATOR

A TOTAL OF 453. EVENTS AND 6472. MINUTES OF WORK WERE GENERATED FOR THE 46 QUERIES

WORK LOAD BY EVENT

EVENT		THIS ITERATION MINUTES	SIMULATION AVERAGE MINUTES
11		45.0	37.0
12		64.0	67.0
13		231.0	253.5
14		46.0	45.5
21		331.0	349.0
22		1055.0	816.5
30		304.0	261.5
31	X	250.0	204.0
32	X	49.0	49.5
33	X	5.0	8.0
40		569.0	568.5
41	X	32.0	31.5
42	X	537.0	537.0
50		1838.0	1796.0
51	X	572.0	457.0
52	X	1266.0	1339.0
60		1408.0	1316.5
61	X	110.0	103.0
62	X	977.0	904.5
63	X	208.0	211.5
64	X	64.0	57.0
65	X	49.0	40.5
72		60.0	68.5
74		168.0	157.5
76		105.0	92.0
81		166.0	154.5
82		19.0	12.0
83		63.0	61.5
TOTAL		6472.0	6057.0

VARIANCE 172225.0

EVENTS MARKED X ARE INTERLOCKED WITH OTHER EVENTS THEIR WORK LOADS ARE INCLUDED IN THEIR MAIN EVENT

PERCENT USE FOR INTERLOCKED EVENTS

EVEN	31	USED	82.2	PERCENT OF THE TIME ATTRIBUTED TO EVENT	30	IN THIS ITERATION	, AND	78.0	PERCENT IN THE SIMULATION
EVENT	32	USED	16.1	PERCENT OF THE TIME ATTRIBUTED TO EVENT	30	IN THIS ITERATION	, AND	18.9	PERCENT IN THE SIMULATION
EVENT	33	USED	1.6	PERCENT OF THE TIME ATTRIBUTED TO EVENT	30	IN THIS ITERATION	, AND	3.0	PERCENT IN THE SIMULATION
EVENT	41	USED	5.6	PERCENT OF THE TIME ATTRIBUTED TO EVENT	40	IN THIS ITERATION	, AND	5.3	PERCENT IN THE SIMULATION
EVENT	42	USED	94.3	PERCENT OF THE TIME ATTRIBUTED TO EVENT	40	IN THIS ITERATION	, AND	94.4	PERCENT IN THE SIMULATION
EVENT	51	USED	31.1	PERCENT OF THE TIME ATTRIBUTED TO EVENT	50	IN THIS ITERATION	, AND	25.4	PERCENT IN THE SIMULATION
EVENT	52	USED	68.8	PERCENT OF THE TIME ATTRIBUTED TO EVENT	50	IN THIS ITERATION	, AND	74.5	PERCENT IN THE SIMULATION
EVENT	61	USED	7.8	PERCENT OF THE TIME ATTRIBUTED TO EVENT	60	IN THIS ITERATION	, AND	7.8	PERCENT IN THE SIMULATION
EVENT	62	USED	69.3	PERCENT OF THE TIME ATTRIBUTED TO EVENT	60	IN THIS ITERATION	, AND	68.7	PERCENT IN THE SIMULATION
EVENT	63	USED	14.7	PERCENT OF THE TIME ATTRIBUTED TO EVENT	60	IN THIS ITERATION	, AND	16.0	PERCENT IN THE SIMULATION
EVENT	64	USED	4.5	PERCENT OF THE TIME ATTRIBUTED TO EVENT	60	IN THIS ITERATION	, AND	4.3	PERCENT IN THE SIMULATION
EVENT	65	USED	3.4	PERCENT OF THE TIME ATTRIBUTED TO EVENT	60	IN THIS ITERATION	, AND	3.0	PERCENT IN THE SIMULATION

QUERY PROCESSING SUMMARY

NUMBER OF QUERIES	COMPLETLY PROCESSED	42
NUMBER OF QUERIES	PARTLY PROCESSED	4
NUMBER OF QUERIES	NOT PROCESSED	0

WORK REMAINING AMOUNTS TO 783. MINUTES

WORK REMAINING BY QUERY GROUP

GROUP	MINUTES	NUMBER
12	180.	1
22	395.	2
31	208.	1

TIME LOST DUE TO MAINTENANCE

EVENT (NUM)	THIS ITER (INTERVALS)	TOTAL MINUTES	AVERAGE MINUTES
11	0	0.	0.
12	0	0.	0.
13	0	0.	0.
14	0	0.	0.
21	0	0.	0.
22	0	0.	0.
30	0	60.	30.
31	0	0.	0.
32	0	0.	0.
33	0	0.	0.
40	0	0.	0.
41	0	0.	0.
42	0	0.	0.
50	2	120.	60.
51	0	0.	0.
52	0	0.	0.
60	1	60.	30.
61	0	0.	0.
62	0	0.	0.
63	0	0.	0.
64	0	0.	0.
65	0	0.	0.
72	1	60.	30.
74	1	60.	30.
76	1	60.	30.
81	0	0.	0.
82	0	0.	0.
83	0	0.	0.

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 420. TO 1860. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	2	1346.5	68.5	1278.0	1377.5	66.0	1311.5	3.0
12	1	1438.2	177.0	1261.2	1327.7	151.0	1176.7	2.0
22	1	2779.5	211.0	2568.5	1807.1	207.3	1599.8	3.0
23	2	1170.7	82.5	1088.2	1260.3	70.0	1190.3	3.[illegible]
31	2	1320.2	184.5	1135.7	1326.1	202.0	1124.1	3.[illegible]
33	2	1695.2	59.0	1636.2	1399.9	71.5	1328.4	4.[illegible]

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO PAGE 6 ITERATION NO. 2

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 1860. TO 3300. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	0	0.0	0.0	0.0	1283.0	77.0	1206.0	1.0
12	0	0.0	0.0	0.0	1109.0	127.5	981.5	2.0
22	2	1184.4	184.0	1000.4	1184.4	184.0	1000.4	2.0
23	0	0.0	0.0	0.0	1124.1	77.0	1047.1	1.0
31	2	2066.5	253.0	1813.5	2066.5	253.0	1813.5	2.0
33	1	1106.9	122.0	984.9	1625.2	102.0	1523.2	3.0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO PAGE 6 ITERATION NO. 2

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 3300. TO 4740. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	0	0.0	0.0	0.0	1743.5	92.3	1651.2	3.0
12	1	975.3	133.0	842.3	1548.6	139.0	1409.6	3.0
22	3	2109.2	234.3	1874.9	1981.2	230.4	1750.8	5.0
23	2	1160.2	71.0	1089.2	1134.0	70.6	1063.3	3.0
31	1	2743.0	274.0	2469.0	2691.8	317.0	2374.8	3.0
33	1	1406.5	66.0	1340.5	1412.5	65.5	1347.0	2.0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO PAGE 6 ITERATION NO. 2

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 4740. TO 6180. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	3	4083.3	81.0	4002.3	3380.9	83.7	3297.1	4.0
12	2	3398.7	142.5	3256.2	4061.5	135.3	3926.2	3.0
22	1	1375.2	166.0	1209.2	1375.2	166.0	1209.2	1.0
23	2	1304.4	67.5	1236.9	2404.8	105.6	2299.2	5.0
31	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	2	3350.6	77.0	3273.6	3727.7	95.8	3631.9	5.0

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO PAGE 6 ITERATION NO. 2

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 6180. TO 7620. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22 X	1	5669.3	386.0	5283.3	5669.3	386.0	5283.3	1.0
23	2	4136.4	111.5	4024.9	4341.1	81.2	4259.9	5.0
31	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	0	0.0	0.0	0.0	4141.3	106.5	4034.8	2.0

QUERY GROUPS FLAGGED WITH AN X CONTAIN INCOMPLETE DATA

INFORMATION RETRIEVAL SYSTEM ALPHA-- SIMULATION NR TWO PAGE 6 ITERATION NO. 2

SUMMARY BY QUERY GROUP

QUERIES ARRIVING FROM 10500. TO 12000. MINUTES

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	0	0.0	0.0	0.0	1370.9	166.0	1204.9	1.0
12 X	1	1374.8	270.0	1104.8	1374.8	270.0	1104.8	1.0
22 X	1	1305.4	238.0	1067.4	1290.8	243.5	1047.3	2.0
23	3	1106.4	65.3	1041.0	1140.4	72.0	1068.4	4.0
31 X	1	799.6	320.0	479.6	799.6	320.0	479.6	1.0
33	4	1103.3	91.2	1012.0	1059.0	88.8	970.2	8.0

QUERY GROUPS FLAGGED WITH AN X CONTAIN INCOMPLETE DATA

SUMMARY BY QUERY GROUP

		ITERATION AVERAGE TIMES			SIMULATION AVERAGE TIMES			
QUERY GROUP	NUM	PROCESS	WORK	DELAY	PROCESS	WORK	DELAY	NUM
11	5	2988.6	76.0	2912.6	2128.4	87.7	2040.6	12.0
12	4	2302.7	148.7	2154.0	2170.4	138.0	2032.4	10.0
22	7	1835.9	206.8	1629.0	1695.6	213.0	1482.6	12.0
23	11	1714.8	78.2	1636.5	2219.0	81.9	2137.0	21.0
31	5	1903.3	229.8	1673.3	2023.3	257.8	1755.5	8.0
33	10	1701.9	32.5	1619.3	2028.9	88.5	1940.3	24.0

SUMMARY OF EVENT UTILIZATION

USE FROM 420. TO 1860. MINUTES

		THIS ITERATION			SIMULATION AVERAGE		
EVENT	UNIT	USE(TIME)	USE(PCT)	DELAY	USE(TIME)	USE(PCT)	DELAY
11	1	6.0	1.2	0.0	6.0	1.2	0.0
12	NQ	24.0	5.0	0.0	21.0	4.3	0.0
13	NQ	46.0	9.5	0.0	38.5	8.0	0.0
14	NQ	0.0	0.0	0.0	0.0	0.0	0.0
21	NQ	68.0	113.3	1399.9	57.0	95.0	1306.6
22	NQ	219.0	365.0	228.0	227.0	378.3	230.0
30	1	70.0	14.5	675.4	57.0	11.8	689.6
40	1	131.0	54.5	526.0	126.5	52.7	443.0
50	1	106.0	29.4	395.0	183.0	50.8	523.0
60	1	185.0	61.6	2446.0	183.5	61.1	2283.5
72	1	5.0	1.0	0.0	8.5	1.7	0.0
	2	0.0	0.0		0.0	0.0	
74	1	36.0	15.0	2105.0	35.5	14.7	1886.0
76	1	23.0	9.5	1.0	17.5	7.2	1.0
81	NQ	34.0	6.2	462.5	31.0	5.7	480.5
82	NQ	1.0	.1	0.0	.5	0.0	0.0
83	NQ	12.0	2.2	369.0	10.0	1.8	372.5

SUMMARY OF EVENT UTILIZATION

USE FROM 1860. TO 3300. MINUTES

		THIS ITERATION			SIMULATION AVERAGE		
EVENT	UNIT	USE(TIME)	USE(PCT)	DELAY	USE(TIME)	USE(PCT)	DELAY
11	1	12.0	2.5	1.7	10.0	2.0	.8
12	NQ	0.0	0.0	0.0	0.0	0.0	0.0
13	NQ	15.0	3.1	0.0	21.5	4.4	0.0
14	NQ	8.0	1.6	1499.0	8.0	1.6	1457.0
21	NQ	0.0	0.0	0.0	0.0	0.0	0.0
22	NQ	220.0	366.6	120.0	117.0	195.0	144.5
30	1	20.0	4.1	1362.0	17.9	3.7	948.9
40	1	66.0	27.5	155.0	88.5	36.8	193.5
50	1	360.0	100.0	645.0	219.0	60.8	397.5
60	1	171.0	57.0	2047.0	186.0	62.0	1945.5
72	1	8.0	1.6	890.4	9.5	1.9	445.2
	2	0.0	0.0		0.0	0.0	
74	1	22.0	9.1	2380.1	15.5	[illegible].4	1787.0
76	1	18.0	7.5	6.0	14.5	6.0	4.0
81	NQ	17.0	3.1	419.9	21.5	3.9	452.0
82	NQ	0.0	0.0	0.0	1.0	.1	0.0
83	NQ	10.0	1.8	278.0	10.5	1.9	347.0

SUMMARY OF EVENT UTILIZATION

USE FROM 3300. TO 4740. MINUTES

EVENT	UNIT	THIS ITERATION USE(TIME)	USE(PCT)	DELAY	SIMULATION AVERAGE USE(TIME)	USE(PCT)	DELAY
11	1	6.0	1.2	0.0	9.0	1.8	0.0
12	NQ	15.0	3.1	0.0	25.0	5.2	0.0
13	NQ	41.0	8.5	0.0	37.5	7.8	0.0
14	NQ	6.0	1.2	940.0	5.5	1.1	931.0
21	NQ	116.0	193.3	623.9	186.5	310.8	969.6
22	NQ	185.0	308.3	114.0	95.0	158.3	113.5
30	1	83.0	17.2	533.6	72.0	15.0	911.6
40	1	50.0	20.8	67.0	116.0	48.3	276.5
50	1	360.0	100.0	1921.0	360.0	100.0	1381.5
60	1	218.0	72.6	2326.0	247.5	82.5	2669.0
72	1	6.0	1.2	0.0	6.0	1.2	0.0
	2	0.0	0.0		0.0	0.0	
7[illegible]	1	28.0	11.6	1476.0	37.5	15.6	2393.3
76	1	10.0	4.1	0.0	18.5	7.7	4.0
81	NQ	32.0	5.9	560.6	30.5	5.6	477.5
82	NQ	5.0	.9	296.0	2.5	.4	148.0
33	NQ	[illegible]	1.1	364.0	5.5	1.2	242.0

H R B - S I N G E R , I N C

SUMMARY OF EVENT UTILIZATION

USE FROM 4740. TO 6180. MINUTES

		THIS ITERATION			SIMULATION AVERAGE		
EVENT	UNIT	USE(TIME)	USE(PCT)	DELAY	USE(TIME)	USE(PCT)	DELAY
11	1	18.0	3.7	0.0	10.5	2.1	0.0
12	NQ	7.0	1.4	0.0	6.5	1.3	0.0
13	NQ	38.0	7.9	0.0	52.5	10.9	0.0
14	NQ	6.0	1.2	1095.0	6.5	1.3	1373.0
21	NQ	3.0	5.0	475.4	2.5	4.1	420.9
22	NQ	210.0	350.0	224.0	240.0	400.0	624.5
30	1	48.7	10.1	1610.8	33.3	6.9	1089.9
40	1	183.0	76.2	288.0	134.0	55.8	575.0
50	1	360.0	100.0	3432.0	360.0	100.0	5071.5
60	1	287.0	95.6	3626.0	240.0	80.0	3174.5
72	1	9.9	2.0	0.0	9.4	1.9	0.0
	2	2.0	.4		3.5	.7	
74	1	35.0	14.5	3256.6	26.0	10.8	2188.4
76	1	30.0	12.5	5.0	21.0	8.7	2.5
81	NQ	35.0	6.4	662.0	27.5	5.0	657.3
82	NQ	0.0	0.0	0.0	1.5	.2	129.5
83	NQ	16.0	2.9	588.0	20.5	3.7	546.5

SUMMARY OF EVENT UTILIZATION

USE FROM 6180. TO 7620. MINUTES

		THIS ITERATION			SIMULATION AVERAGE		
EVENT	UNIT	USE(TIME)	USE(PCT)	DELAY	USE(TIME)	USE(PCT)	DELAY
11	1	3.0	.6	0.0	1.5	.3	0.0
12	NQ	0.0	0.0	0.0	0.0	0.0	0.0
13	NQ	23.0	4.7	0.0	42.5	8.8	0.0
14	NQ	9.0	1.8	1138.0	9.0	1.8	988.0
21	NQ	0.0	0.0	0.0	0.0	0.0	0.0
22	NQ	0.0	0.0	0.0	0.0	0.0	0.0
30	1	7.0	1.4	305.1	13.0	2.7	534.5
40	1	31.0	51.6	707.0	18.0	30.0	353.5
50	1	0.0	0.0	0.0	0.0	0.0	0.0
60	1	104.0	86.6	842.0	112.0	93.3	1422.0
72	1	9.0	1.8	0.0	10.0	2.0	0.0
	2	0.0	0.0		2.0	.4	
74	1	9.0	3.7	474.0	11.5	4.7	700.3
76	1	3.0	1.2	0.0	7.0	2.9	1.5
81	NQ	10.0	2.0	444.1	13.5	2.8	586.1
82	NQ	3.0	.6	0.0	1.5	.3	0.0
83	NQ	5.0	1.0	0.0	2.5	.5	0.0

SUMMARY OF EVENT UTILIZATION

USE FROM 10500. TO 12000. MINUTES

		THIS ITERATION			SIMULATION AVERAGE		
EVENT	UNIT	USE(TIME)	USE(PCT)	DELAY	USE(TIME)	USE(PCT)	DELAY
11	1	0.0	0.0	0.0	0.0	0.0	0.0
12	NQ	18.0	3.7	0.0	14.5	3.0	0.0
13	NQ	68.0	14.1	0.0	61.0	12.7	0.0
14	NQ	13.0	2.4	3585.0	14.5	2.6	3472.5
21	NQ	144.0	240.0	1220.9	103.0	171.6	1027.6
22	NQ	7.0	11.6	3747.0	30.5	50.8	1911.0
30	1	63.3	13.1	3979.9	62.1	12.9	2145.7
40	1	13.0	5.4	28.0	38.0	15.8	43.5
50	1	208.0	57.7	695.0	272.0	75.5	2082.0
60	1	300.0	100.0	4051.0	255.5	85.1	3235.0
72	1	12.0	2.5	0.0	11.5	2.3	0.0
	2	8.0	1.6		8.0	1.6	
74	1	21.0	8.7	1045.0	23.0	9.5	927.0
76	1	16.0	6.6	3.0	11.0	4.5	1.5
81	NQ	29.0	5.3	951.5	26.0	4.8	586.0
82	NQ	6.0	1.1	306.0	3.0	.5	153.0
83	NQ	0.0	0.0	0.0	4.5	.8	62.0

SUMMARY OF EVENT UTILIZATION

EVENT	UNIT	THIS ITERATION USE(TIME)	USE(PCT)	DELAY	SIMULATION AVERAGE USE(TIME)	USE(PCT)	DELAY
11	1	45.0	1.5	1.7	37.0	1.2	.8
12	NQ	64.0	2.2	0.0	67.0	2.3	0.0
13	NQ	231.0	8.0	0.0	253.5	8.6	0.0
14	NQ	42.0	1.4	8257.0	43.5	1.4	8221.5
21	NQ	331.0	91.9	3720.2	349.0	96.9	3724.9
22	NQ	841.0	200.2	4433.0	709.5	168.9	3023.5
30	1	292.1	10.1	8467.1	255.5	8.8	6370.3
40	1	474.0	29.2	1771.0	521.0	32.1	1875.0
50	1	1568.0	62.2	25487.0	1661.0	65.9	33009.0
60	1	1265.0	63.8	15338.0	1245.0	62.8	16166.5
72	1	49.9	1.7	890.4	54.9	1.9	445.2
	2	10.0	.3		13.5	.4	
74	1	151.0	10.4	10736.9	149.0	10.3	9882.2
76	1	100.0	6.9	15.0	89.5	6.2	14.5
81	NQ	157.0	4.7	3500.8	150.0	4.5	3239.7
82	NQ	15.0	.4	602.0	10.0	.3	430.5
83	NQ	51.0	1.5	1861.0	55.5	1.6	1701.0

APPENDIX C

GENERAL LOGIC CHARTS

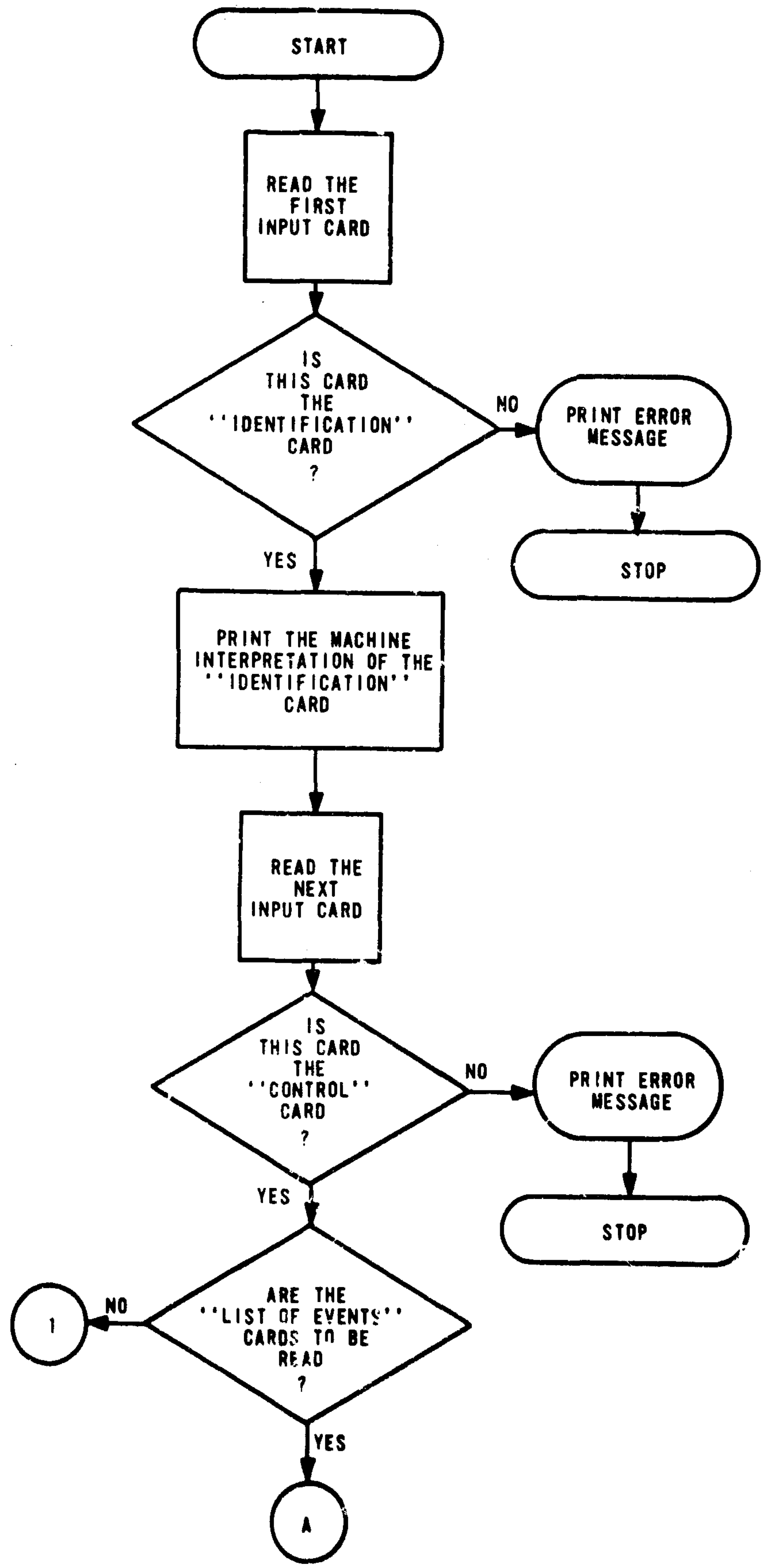


FIG. 14 INPUT LOGIC

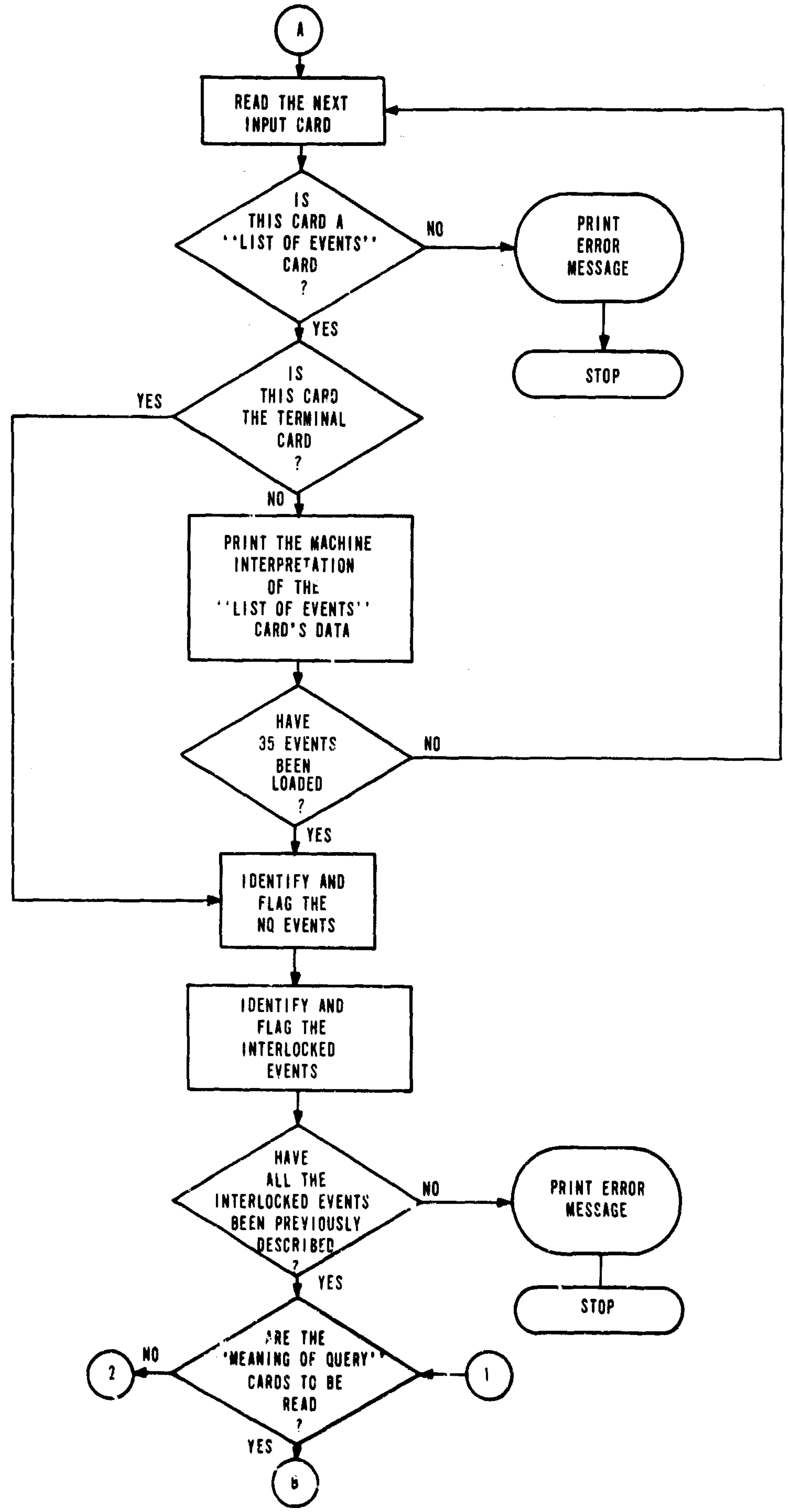


FIG. 14 INPUT LOGIC (CONT'D)

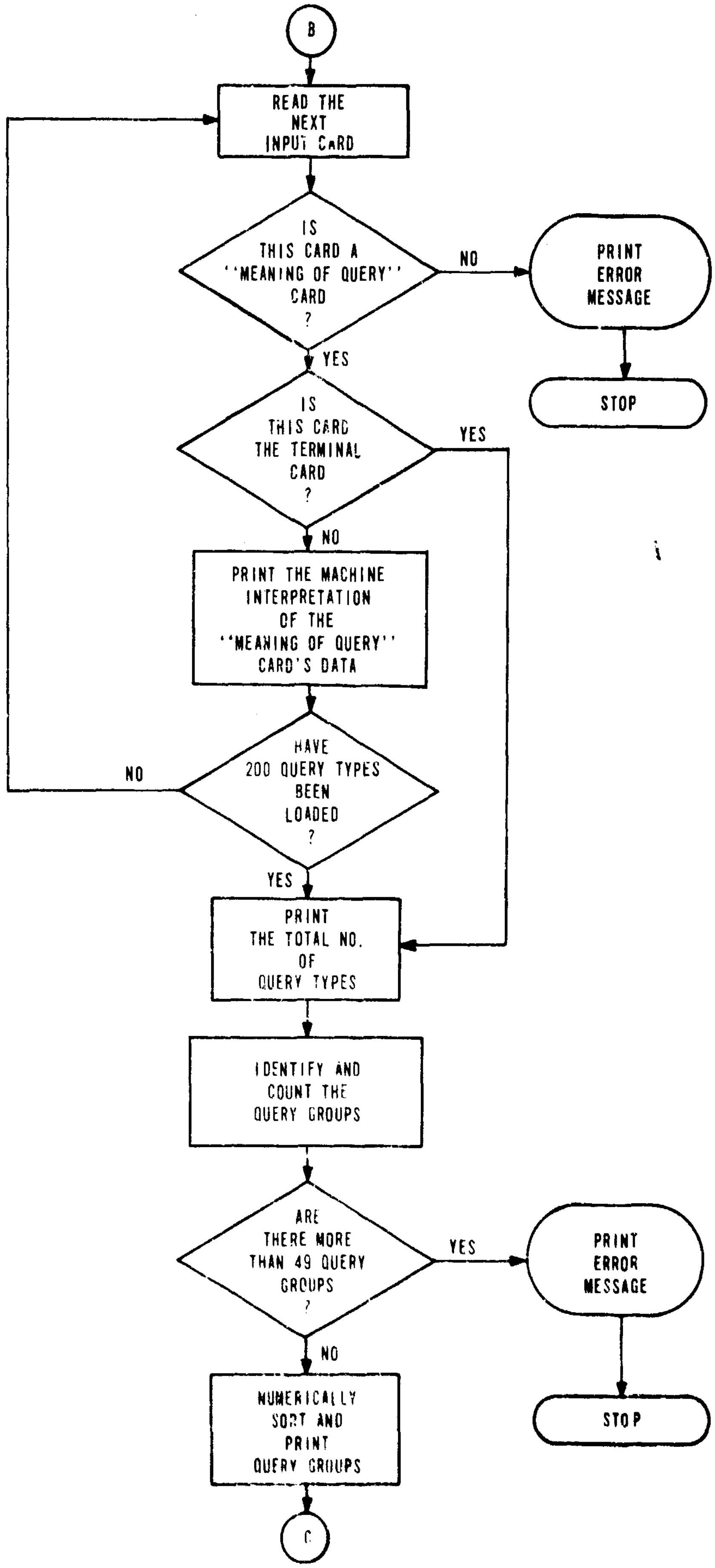


FIG 14 INPUT LOGIC (CONT'D)

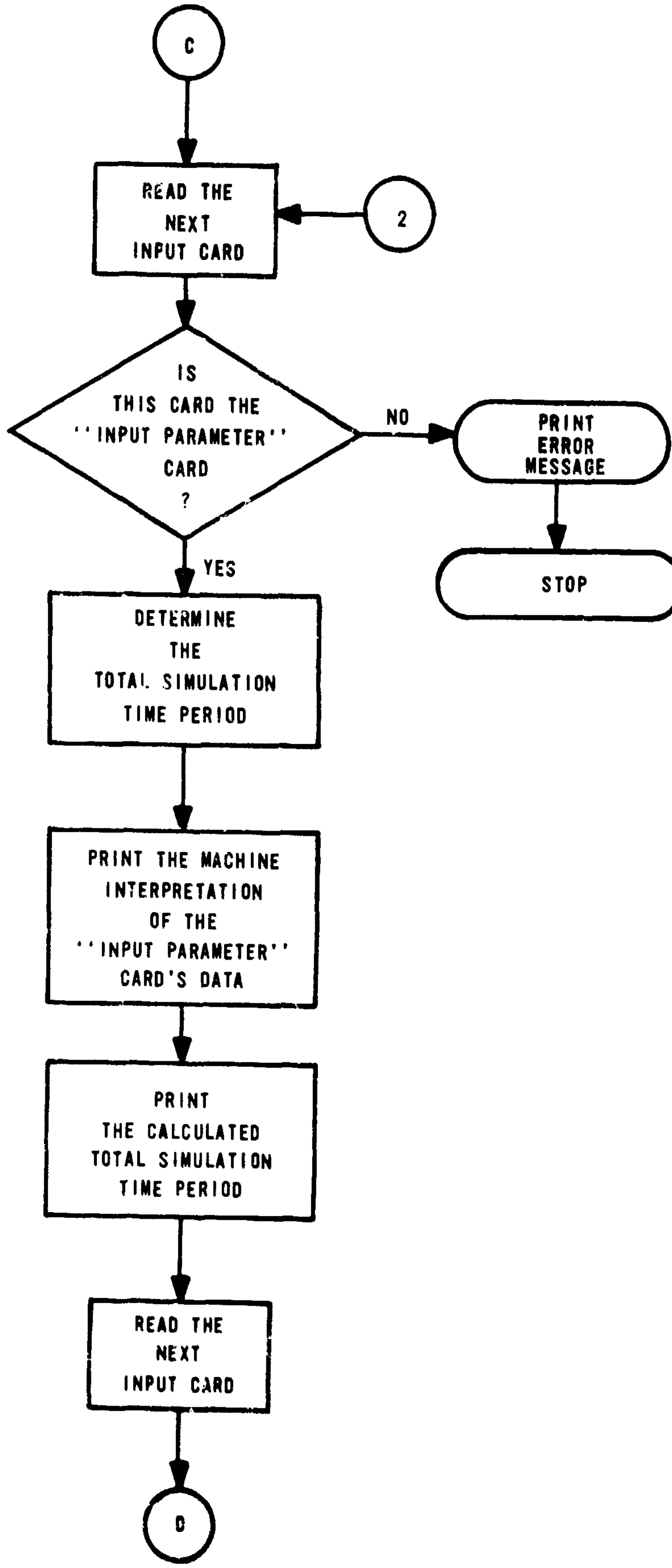


FIG. 14 INPUT LOGIC (CONT'D)

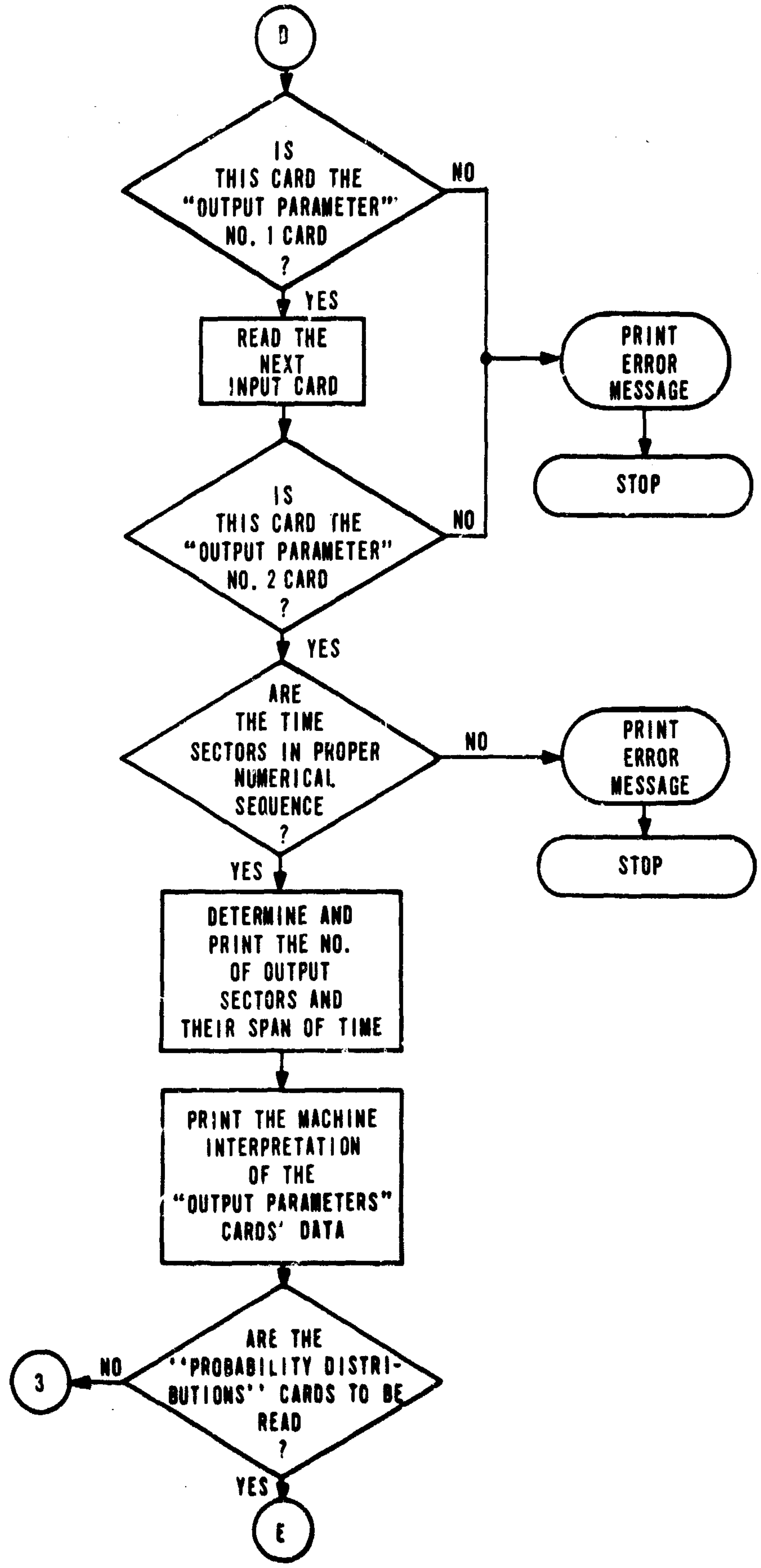


FIG. 14 INPUT LOGIC (CONT'D)

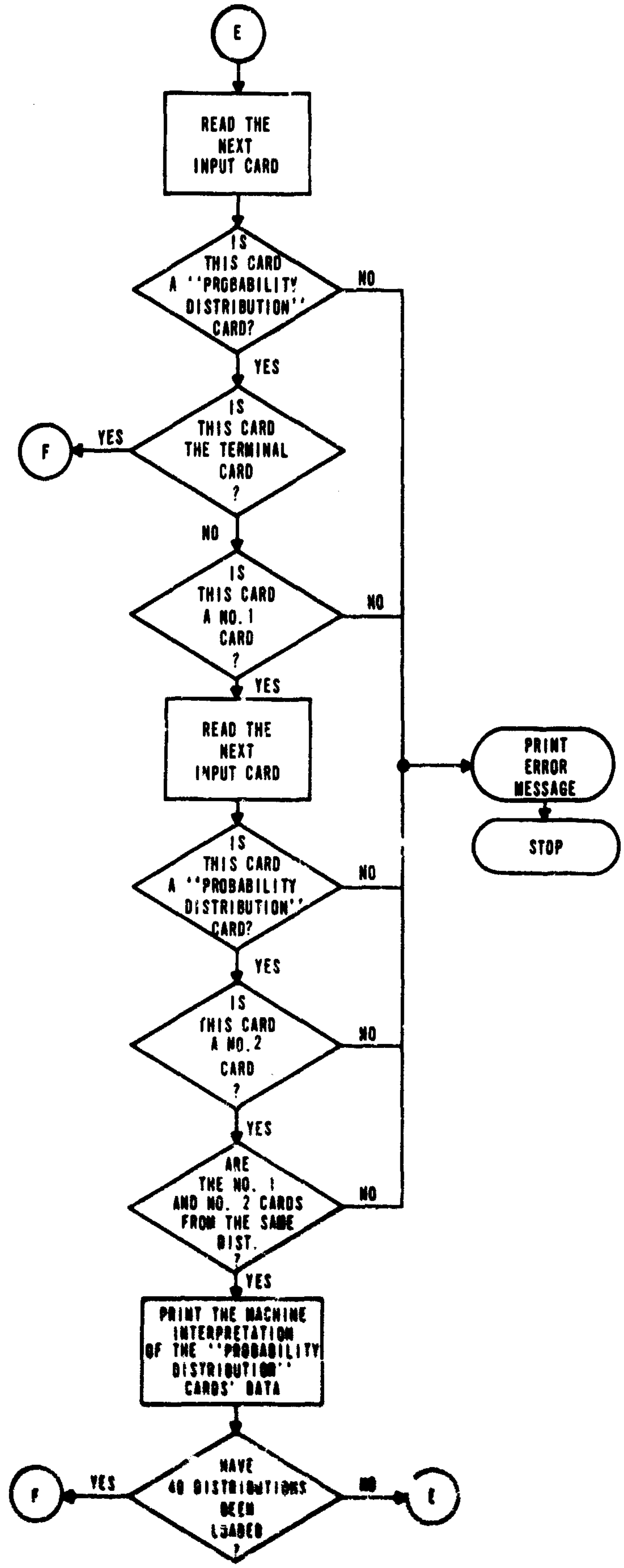


FIG. 14 INPUT LOGIC (CONT'D)

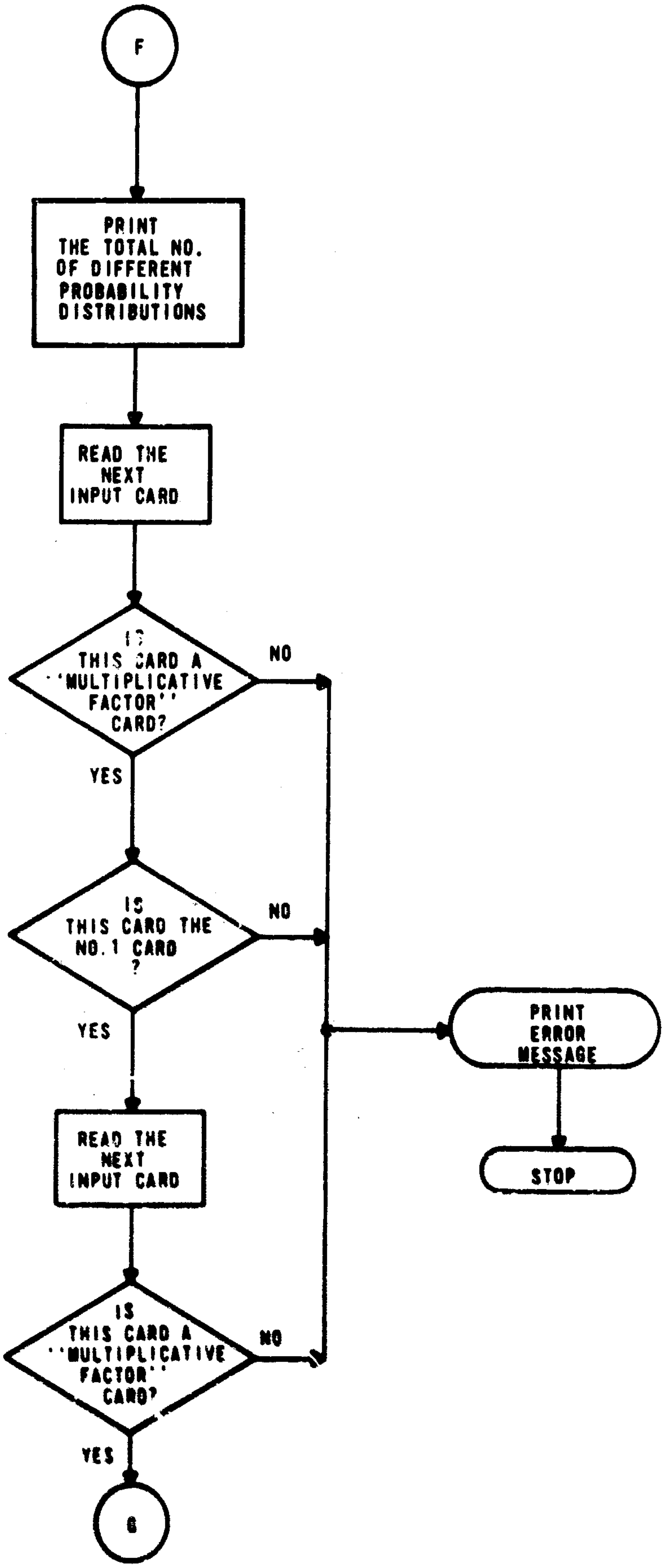


FIG. 14 INPUT LOGIC (CONT'D)

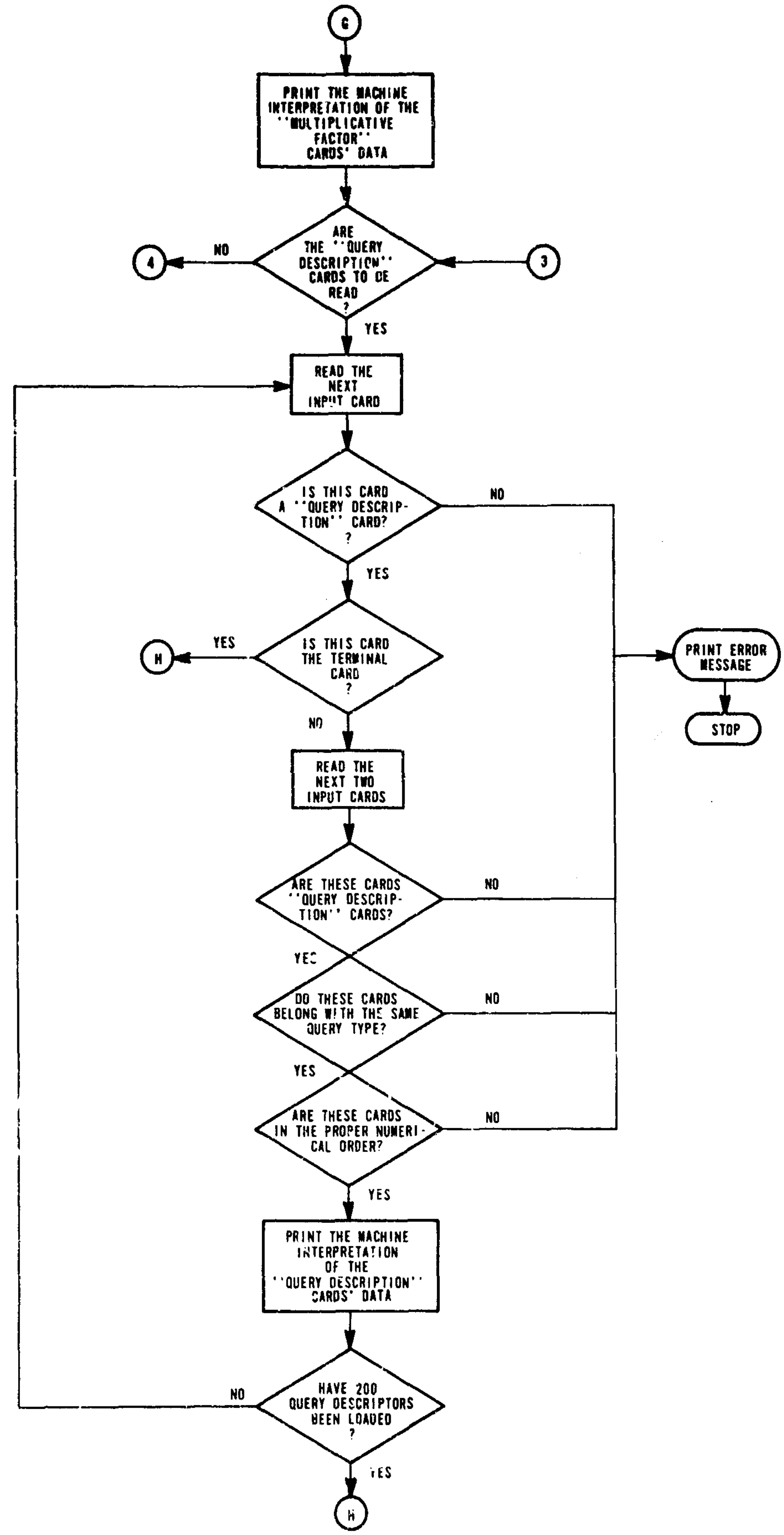


FIG. 14 INPUT LOGIC (CONT'D)

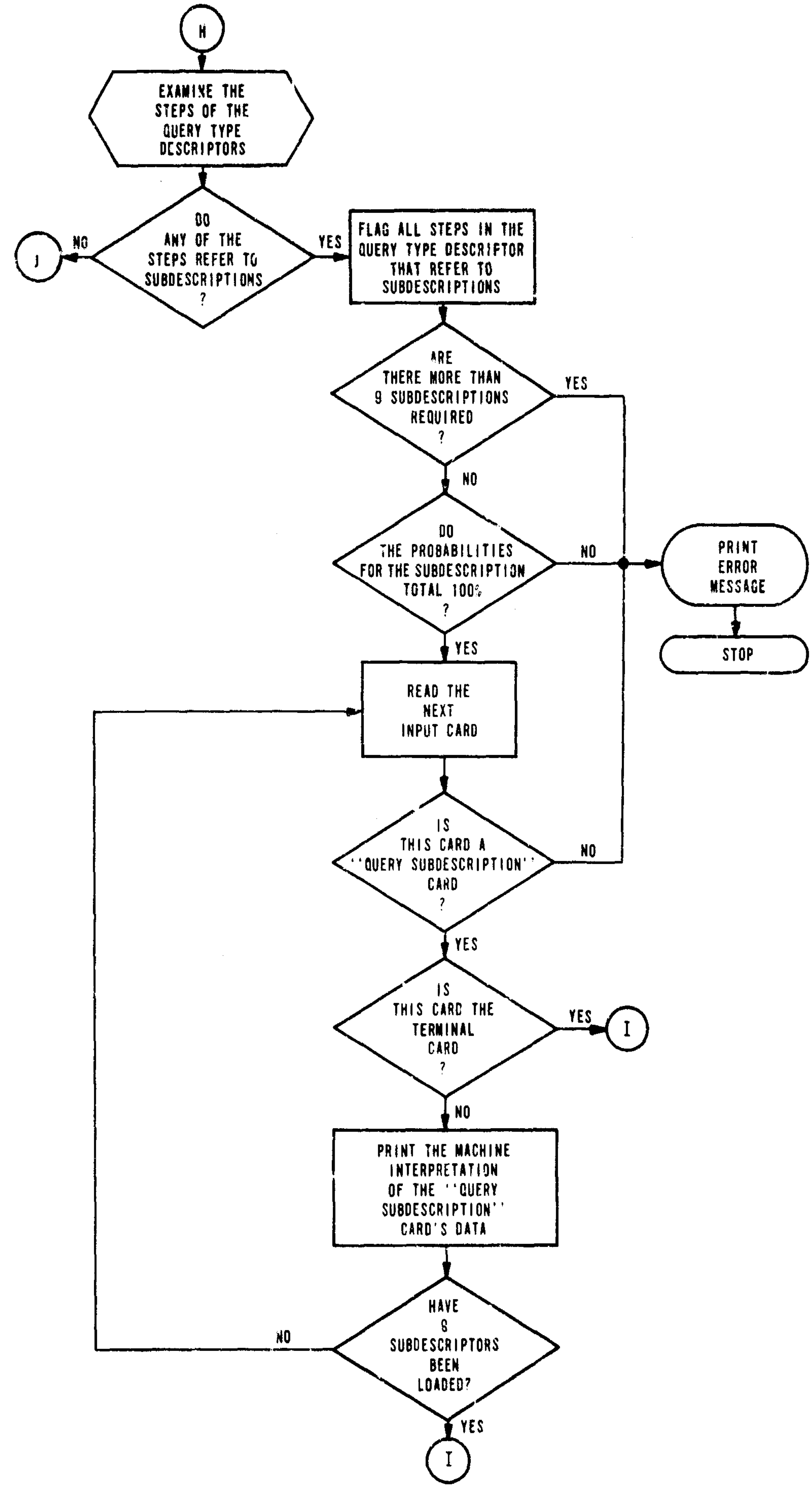


FIG. 14 INPUT LOGIC (CONT'D)

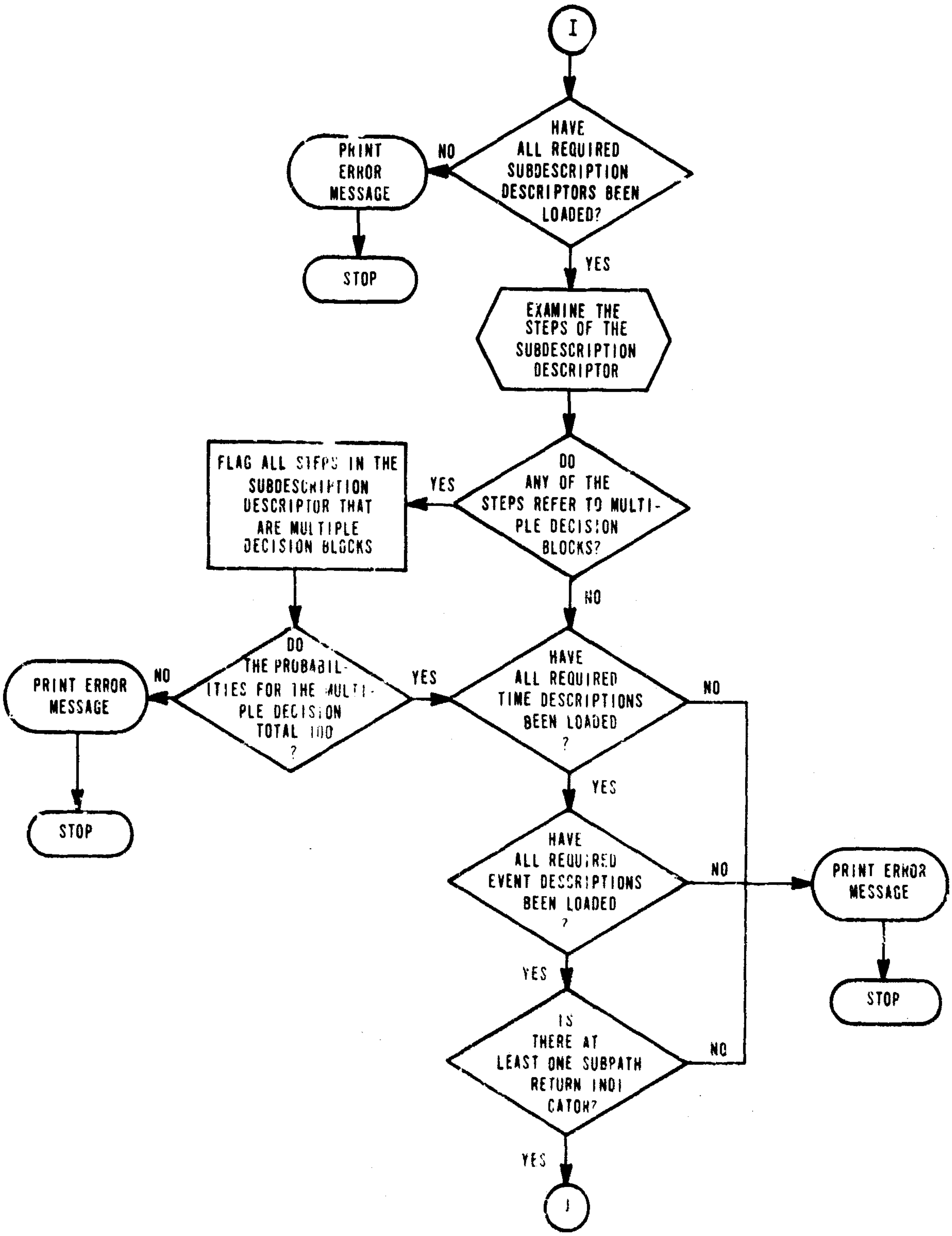


FIG 14 INPUT LOGIC (CONT D)

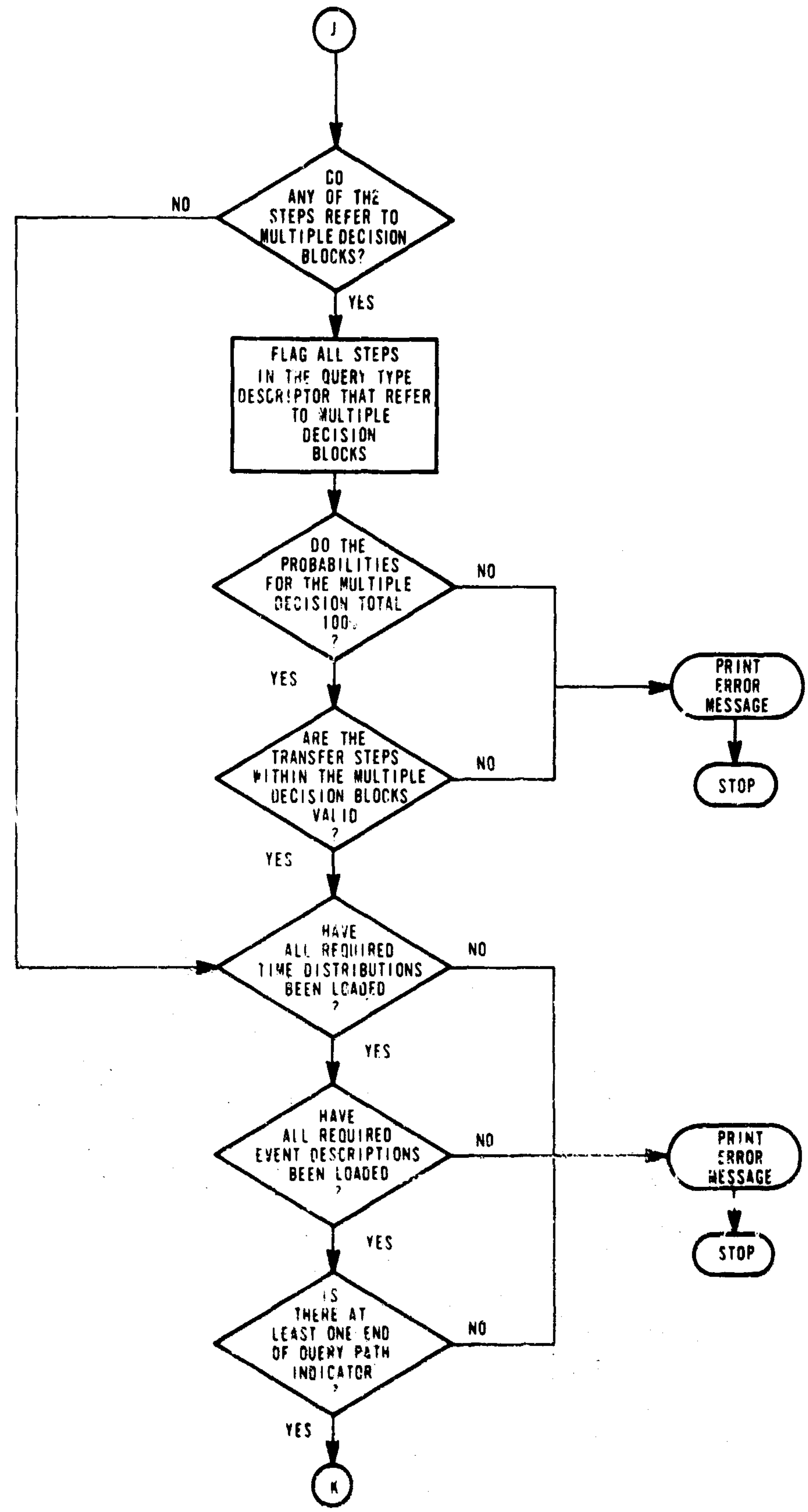


FIG. 14 INPUT LOGIC (CONT'D)

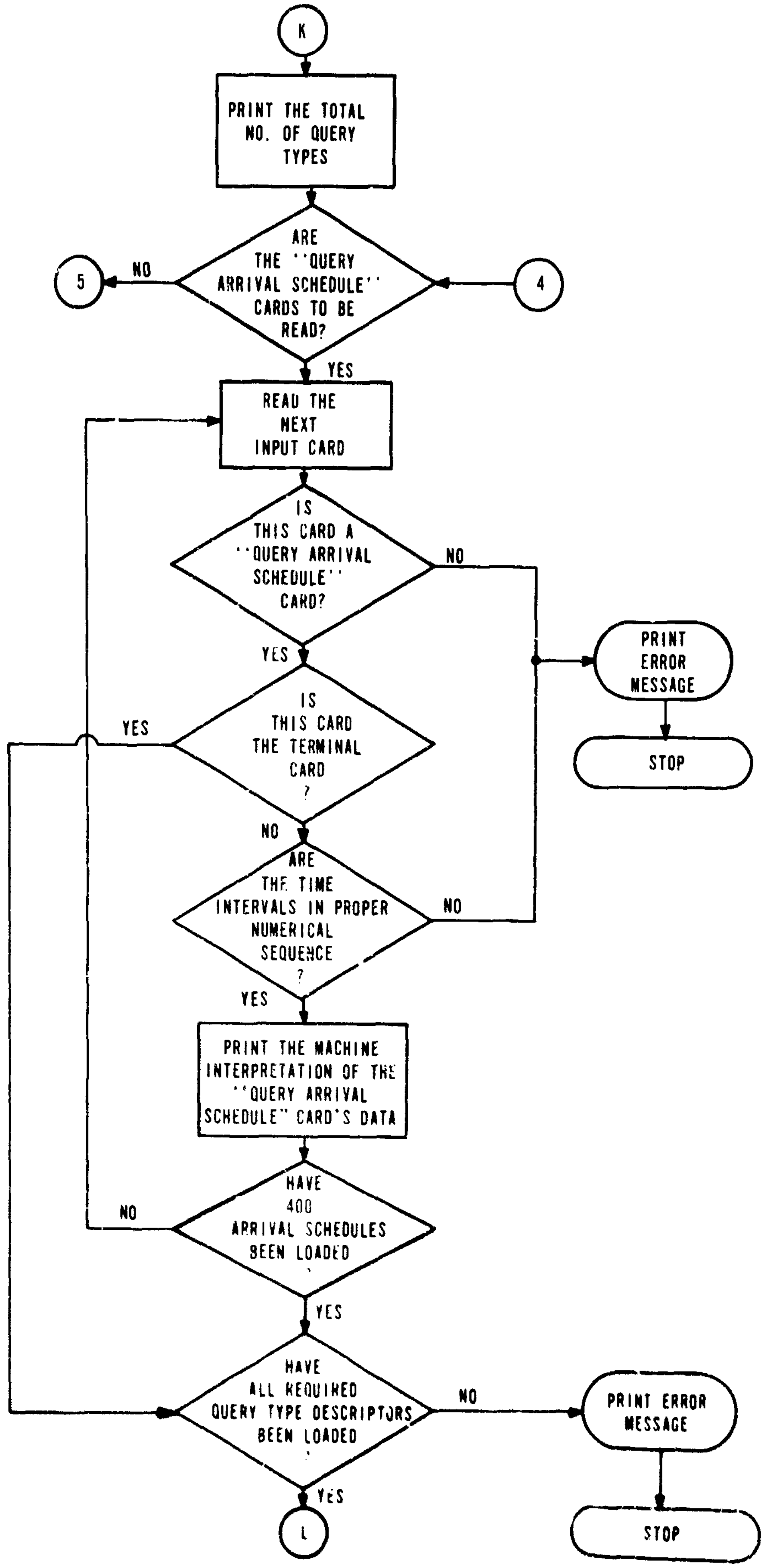


FIG. 14 INPUT LOGIC (CONT'D)

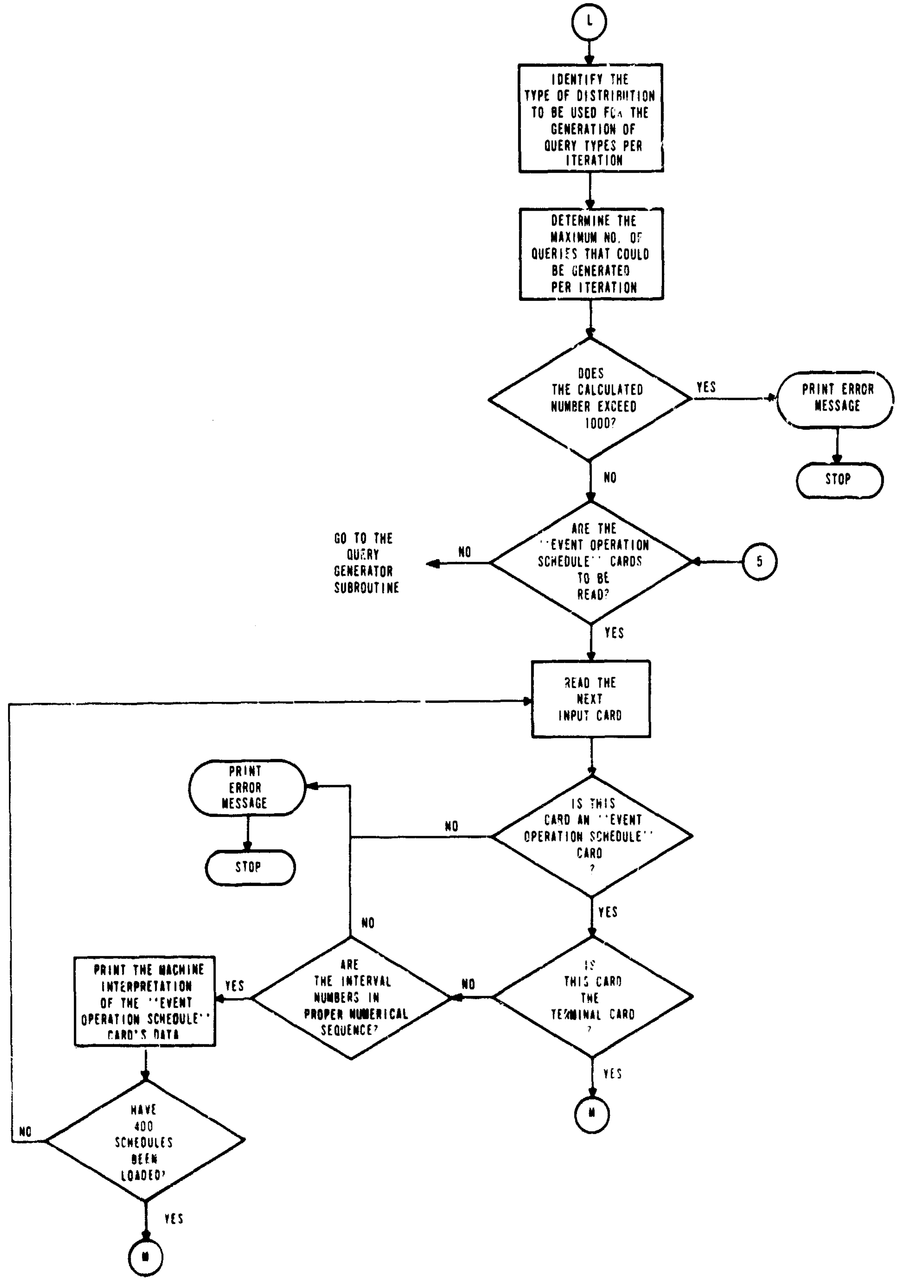


FIG 14 INPUT LOGIC (CONT'D)

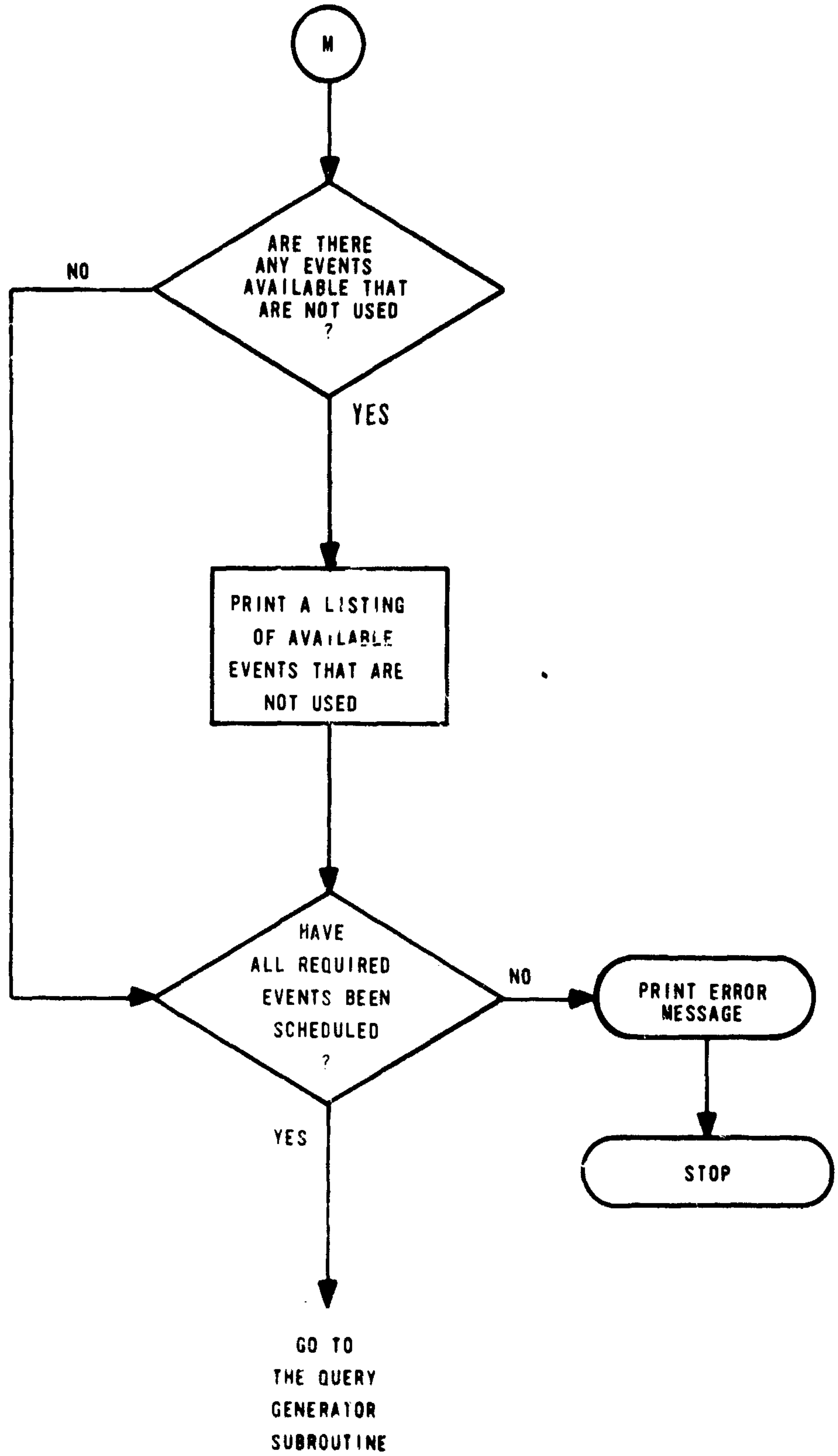


FIG. 14 INPUT LOGIC (CONT'D)

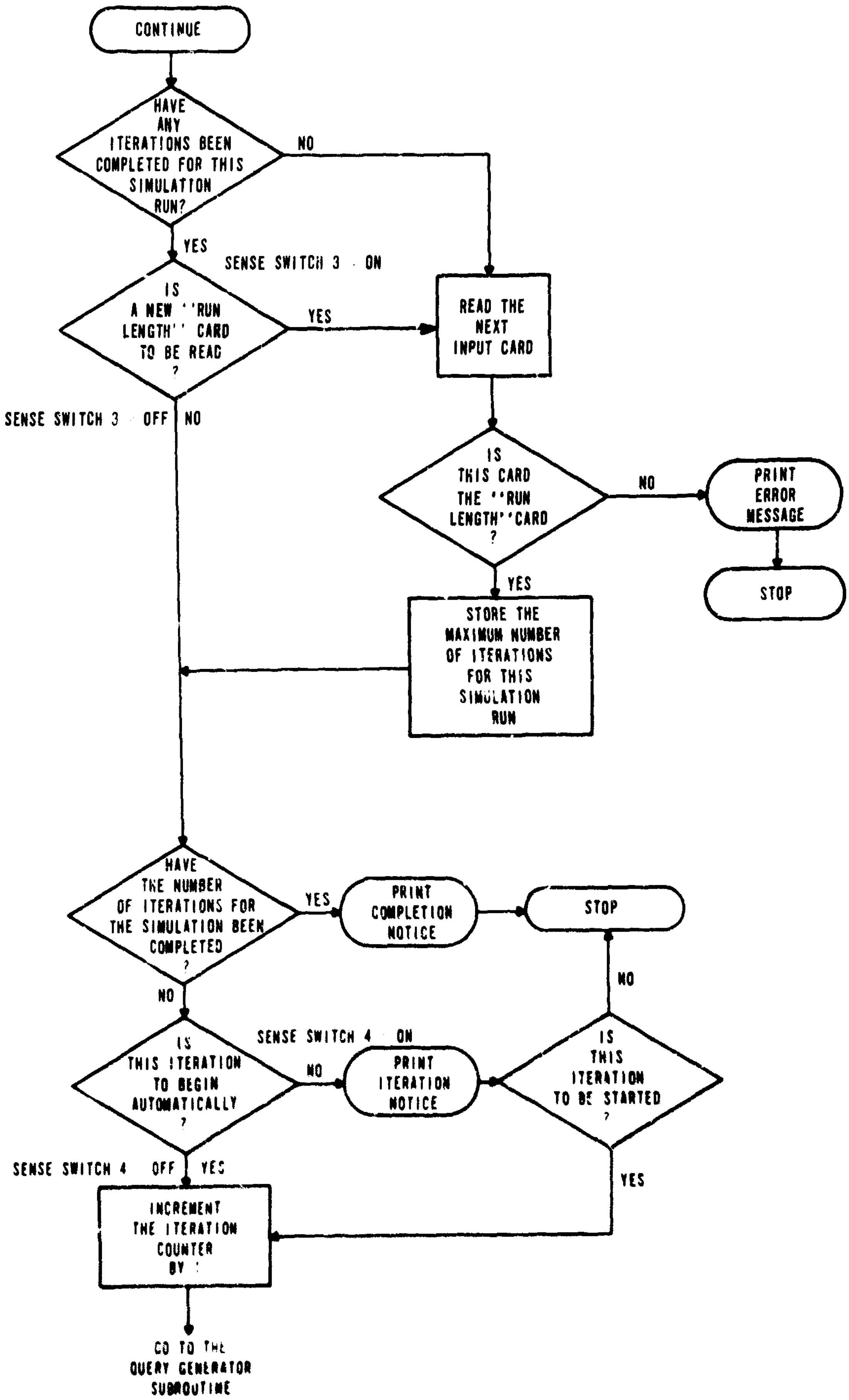


FIG. 15 RESTART OR REITERATE LOGIC

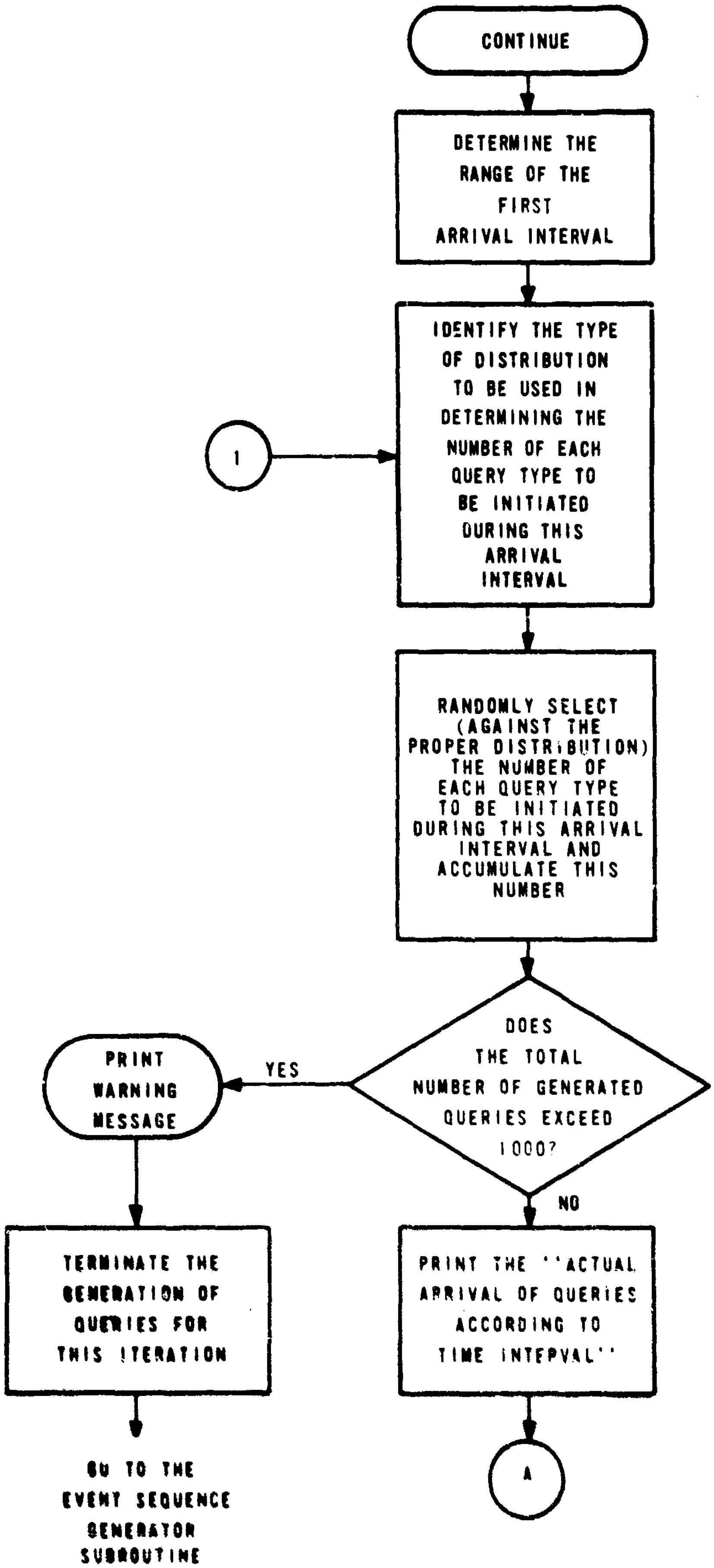


FIG. 16 QUERY GENERATOR LOGIC

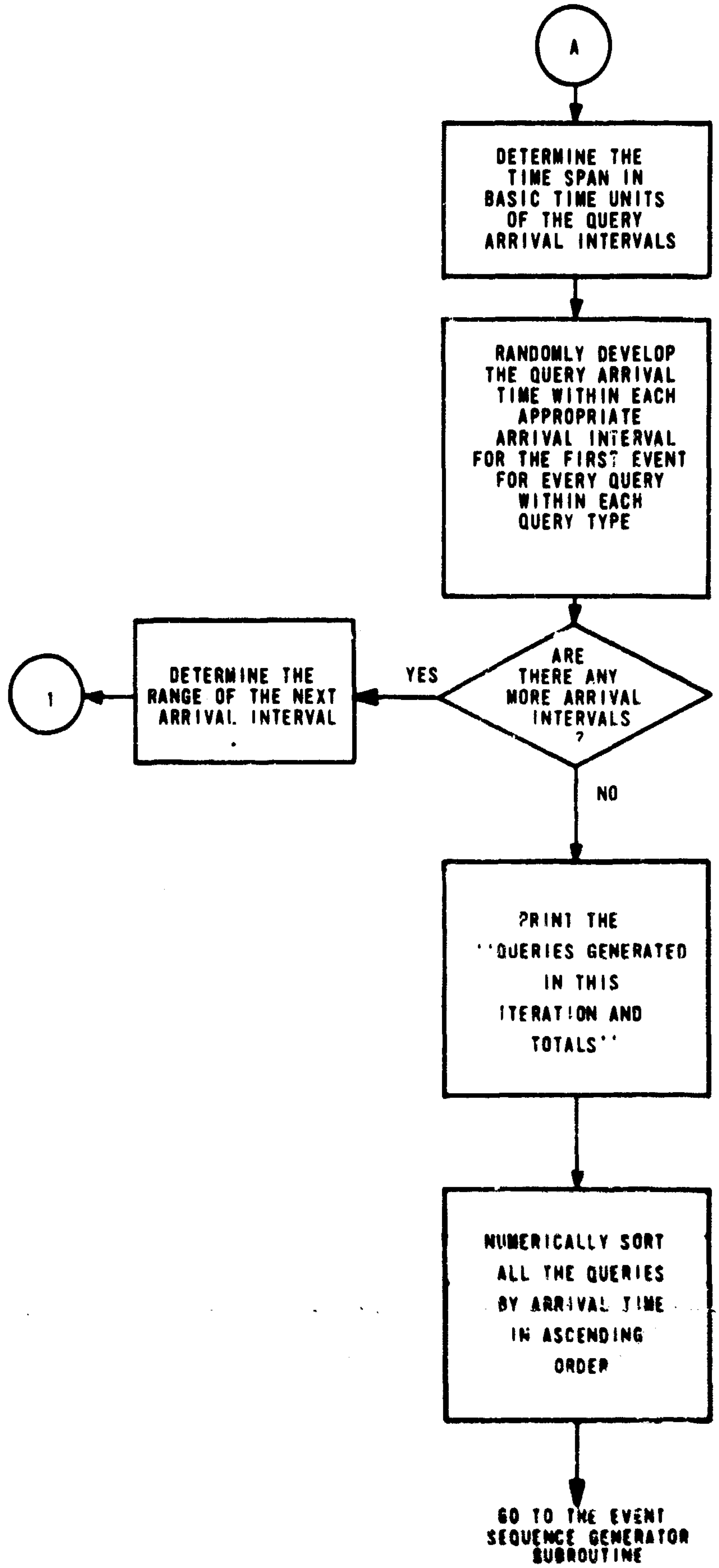


FIG. 16 QUERY GENERATOR LOGIC (CONT'D)

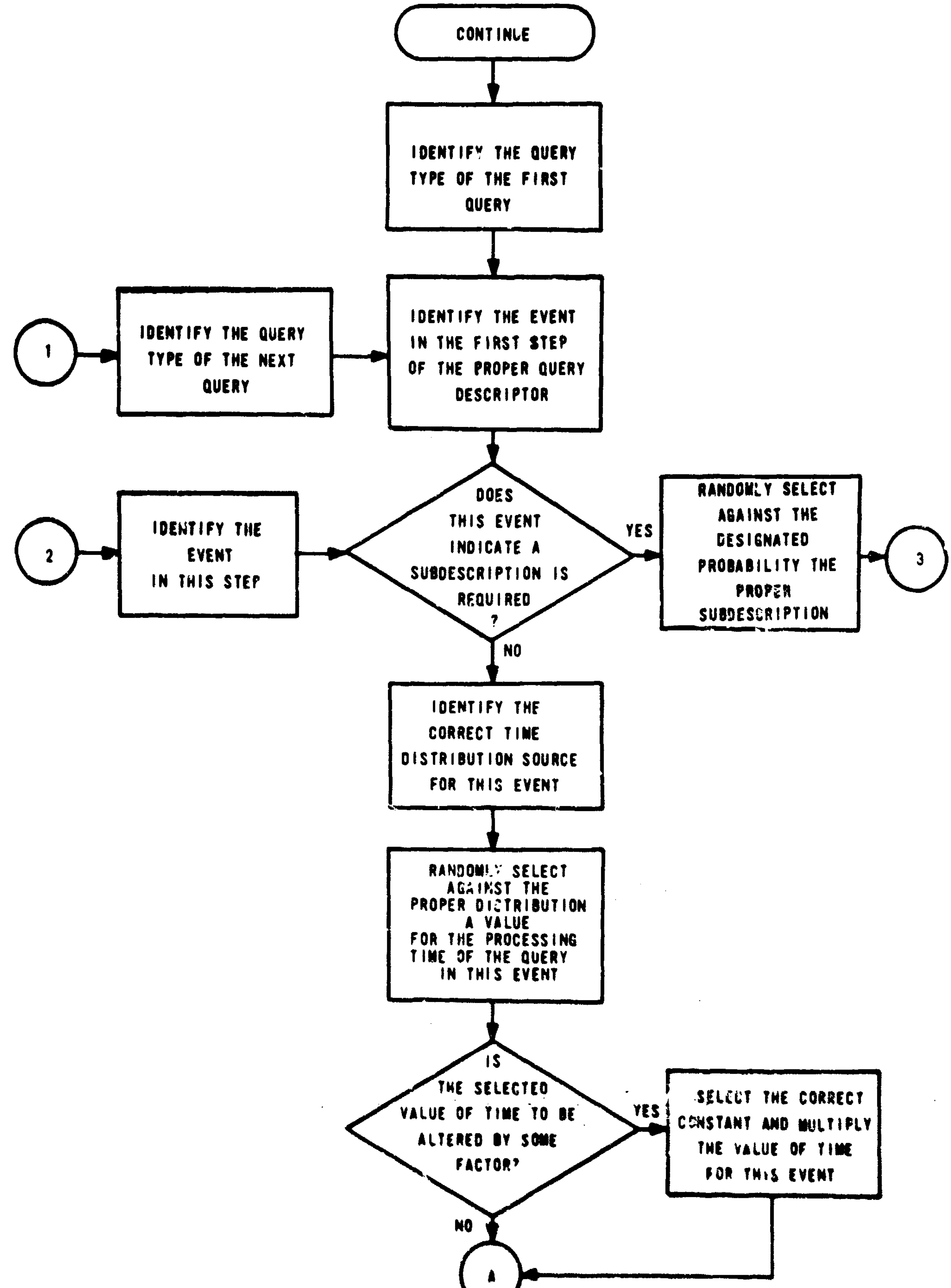


FIG. 17 EVENT SEQUENCE GENERATOR LOGIC

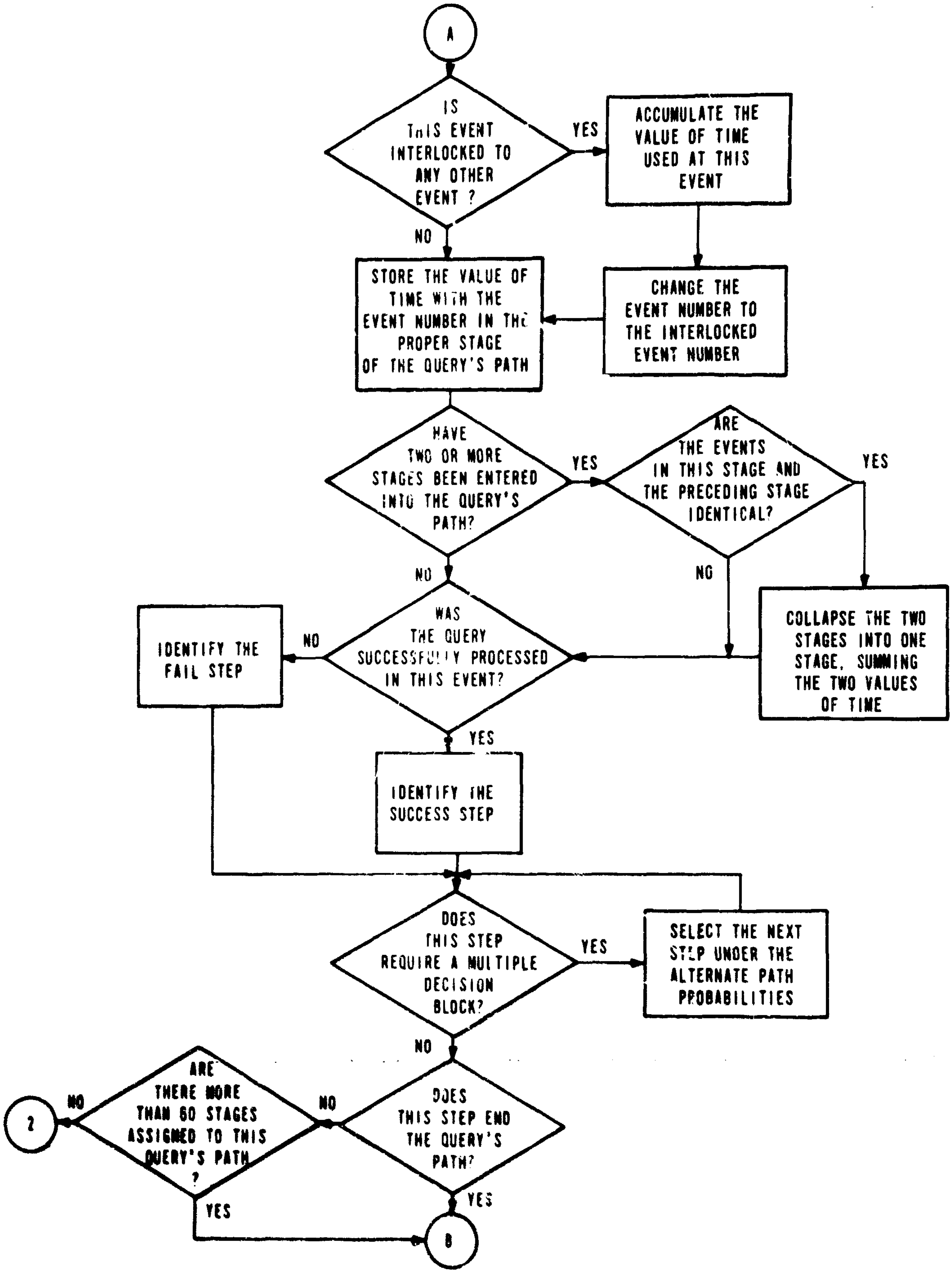


FIG. 17 EVENT SEQUENCE GENERATOR LOGIC (CONT'D)

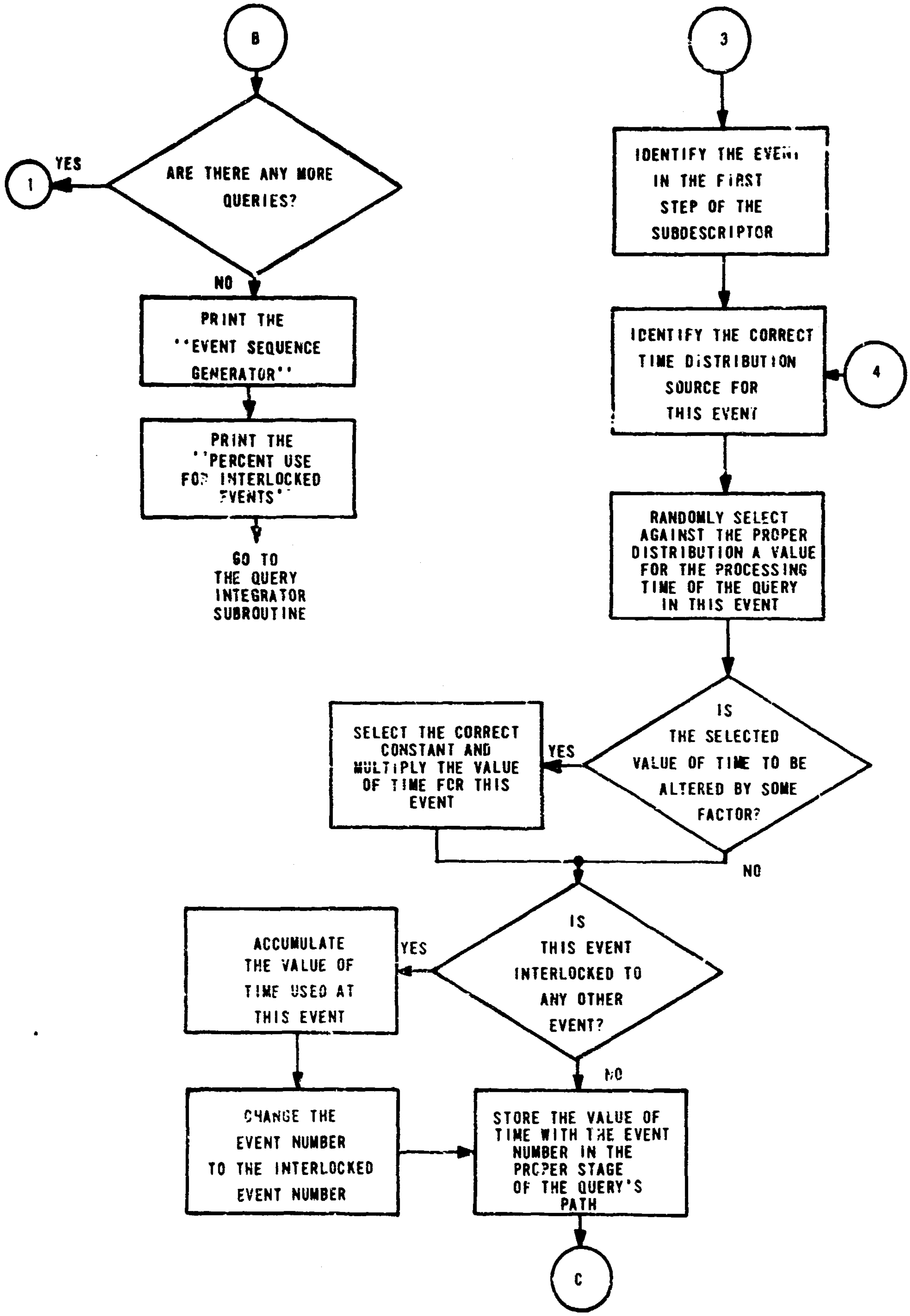


FIG. 17 EVENT SEQUENCE GENERATOR LOGIC (CONT'D)

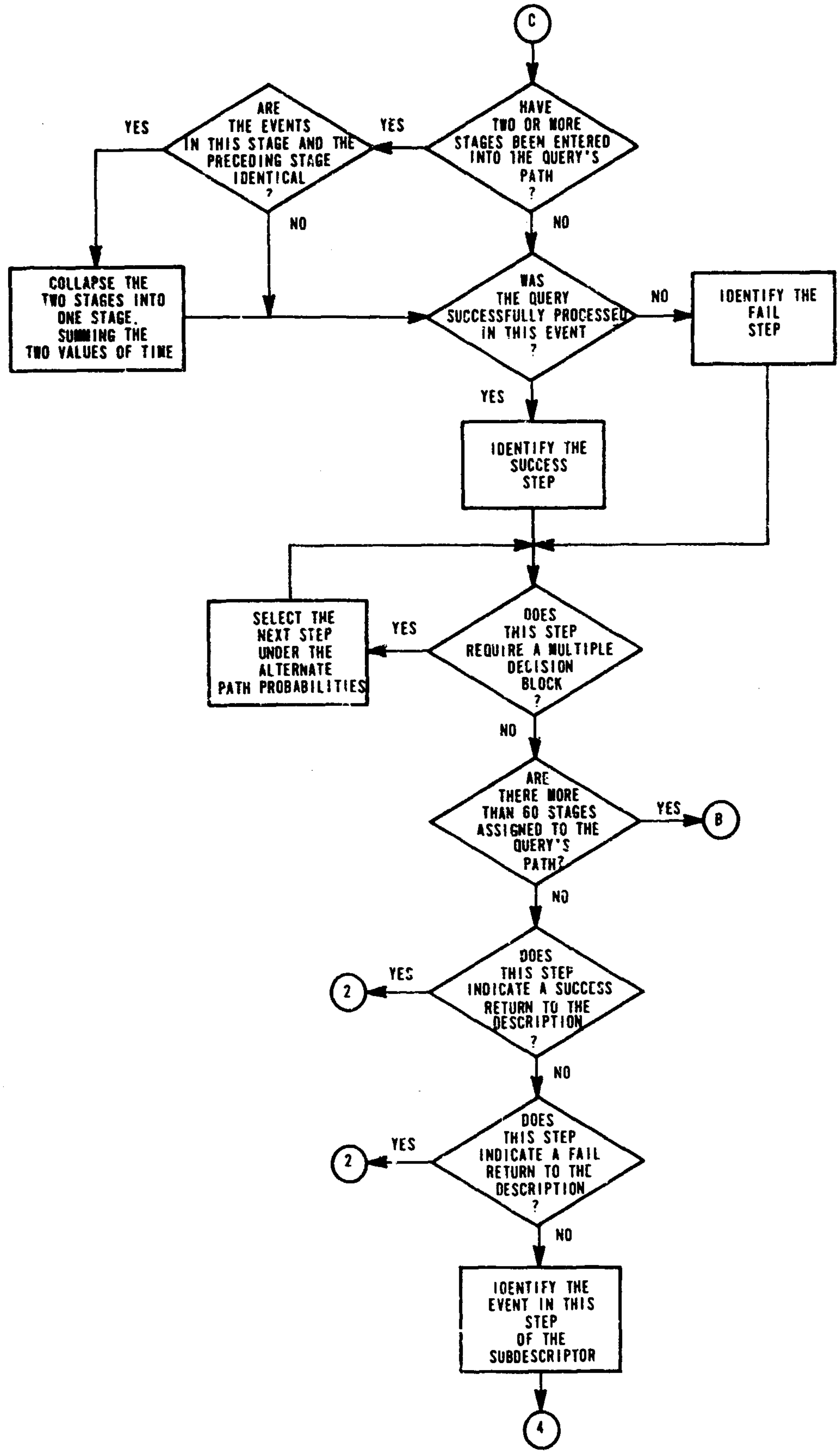


FIG. 17 EVENT SEQUENCE GENERATOR LOGIC (CONT'D)

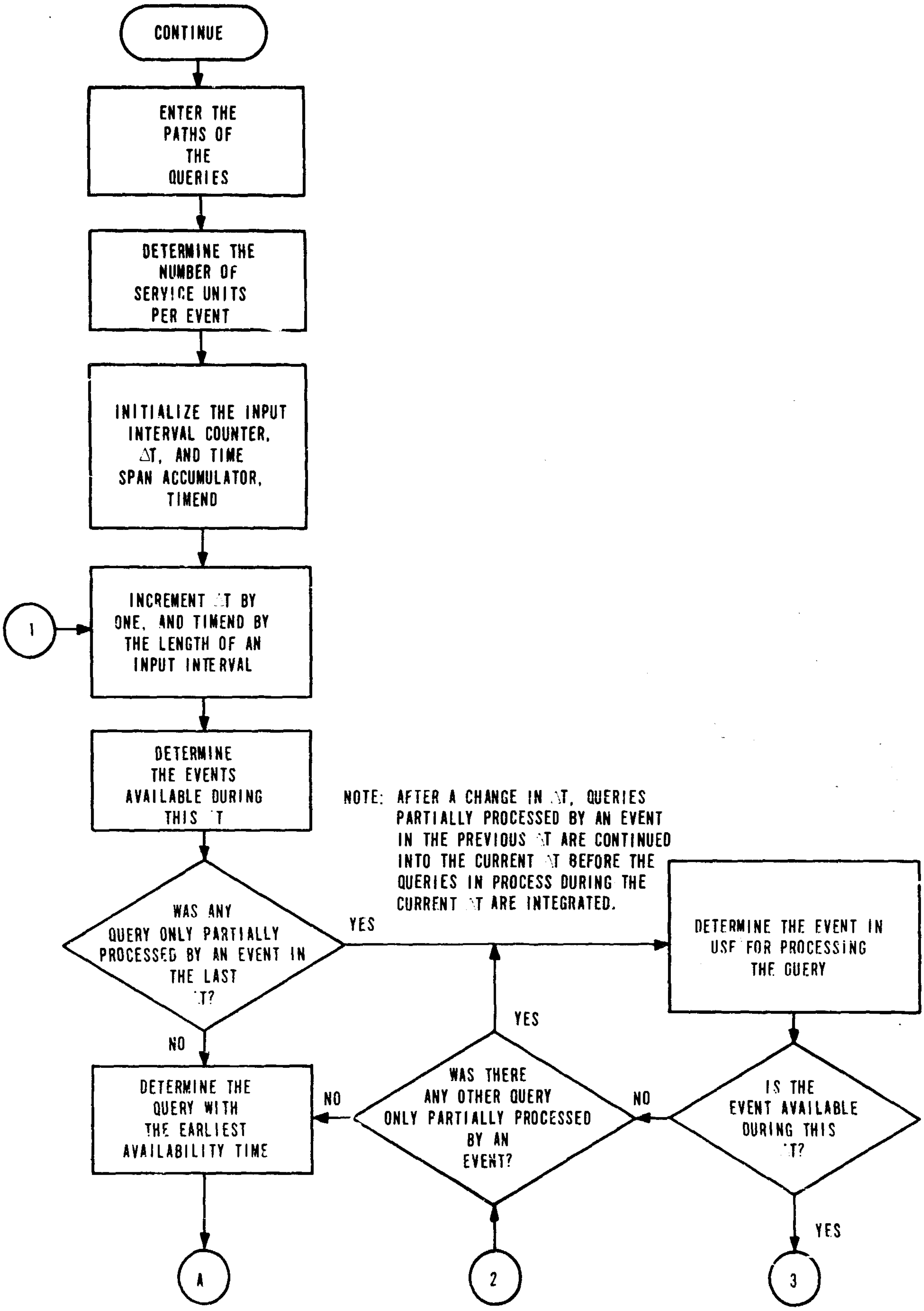
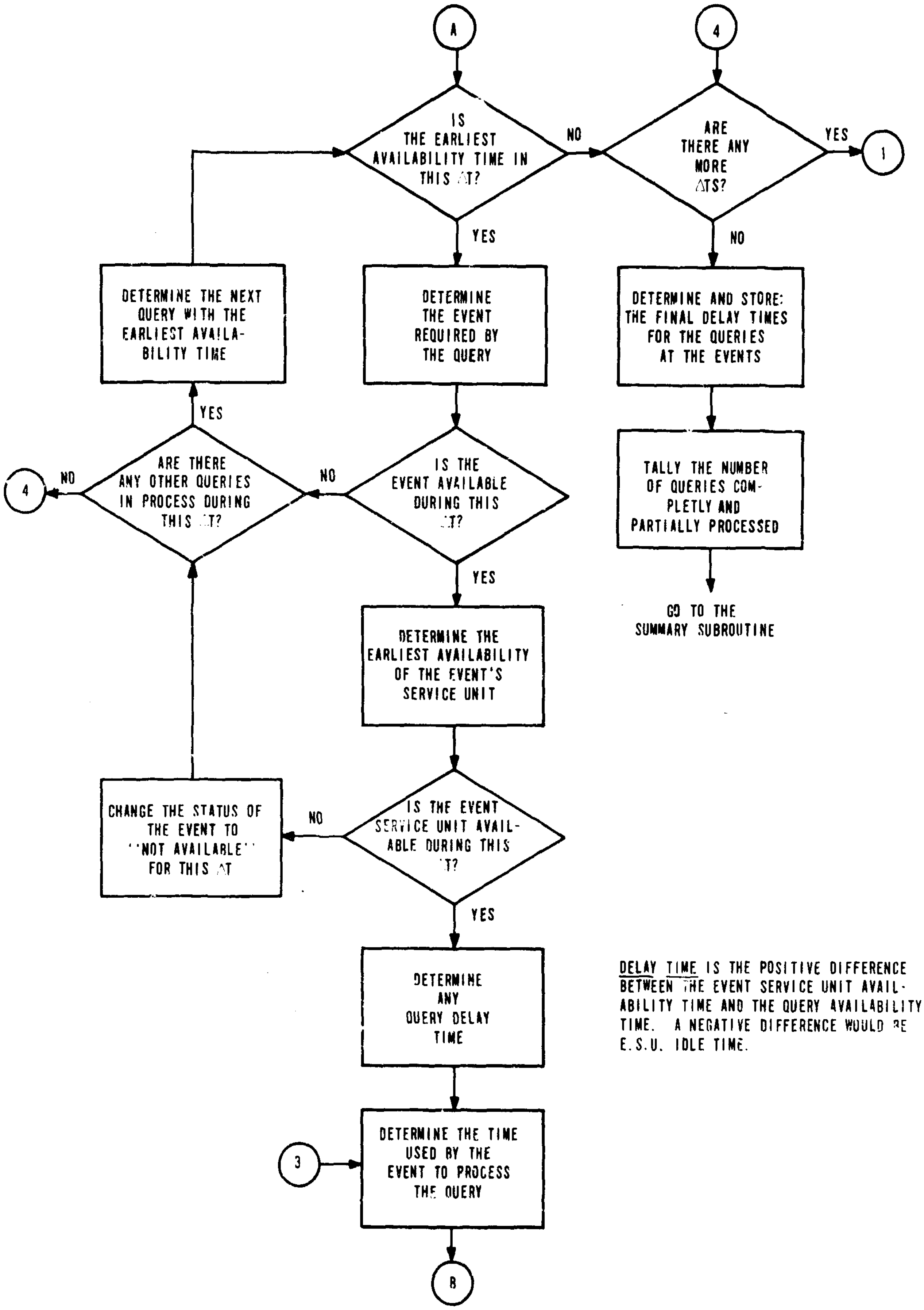


FIG. 18 QUERY INTEGRATOR LOGIC



DELAY TIME IS THE POSITIVE DIFFERENCE BETWEEN THE EVENT SERVICE UNIT AVAIL-ABILITY TIME AND THE QUERY AVAILABILITY TIME. A NEGATIVE DIFFERENCE WOULD BE E.S.U. IDLE TIME.

FIG. 18 QUERY INTEGRATOR LOGIC (CONT'D)

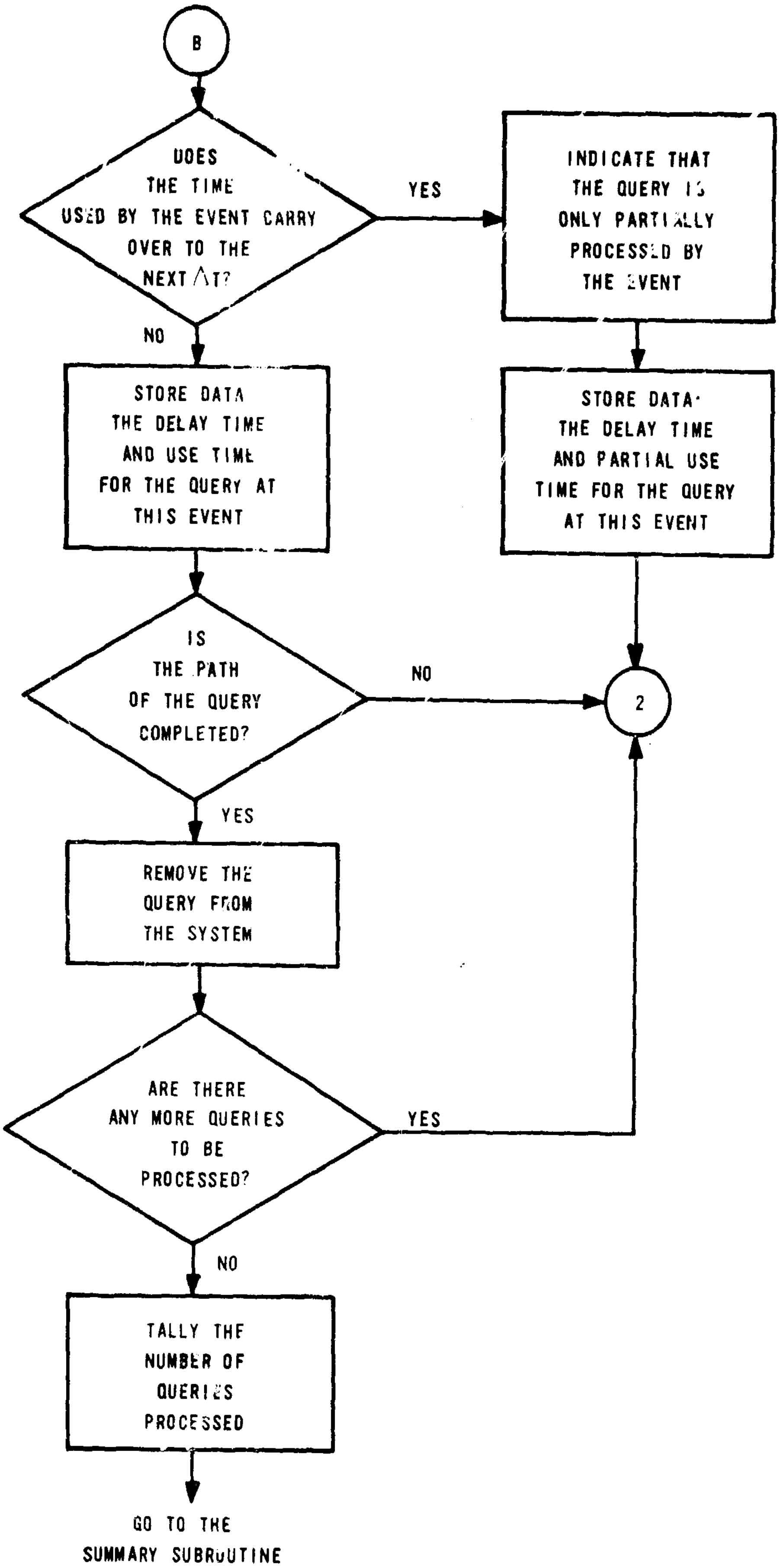


FIG. 18 QUERY INTEGRATOR LOGIC (CONT'D)

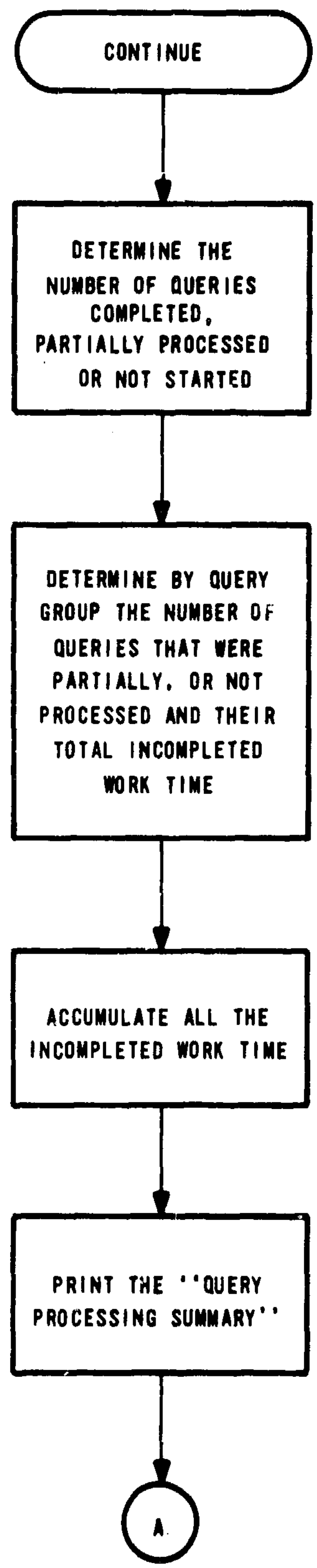


FIG. 19 SUMMARY LOGIC

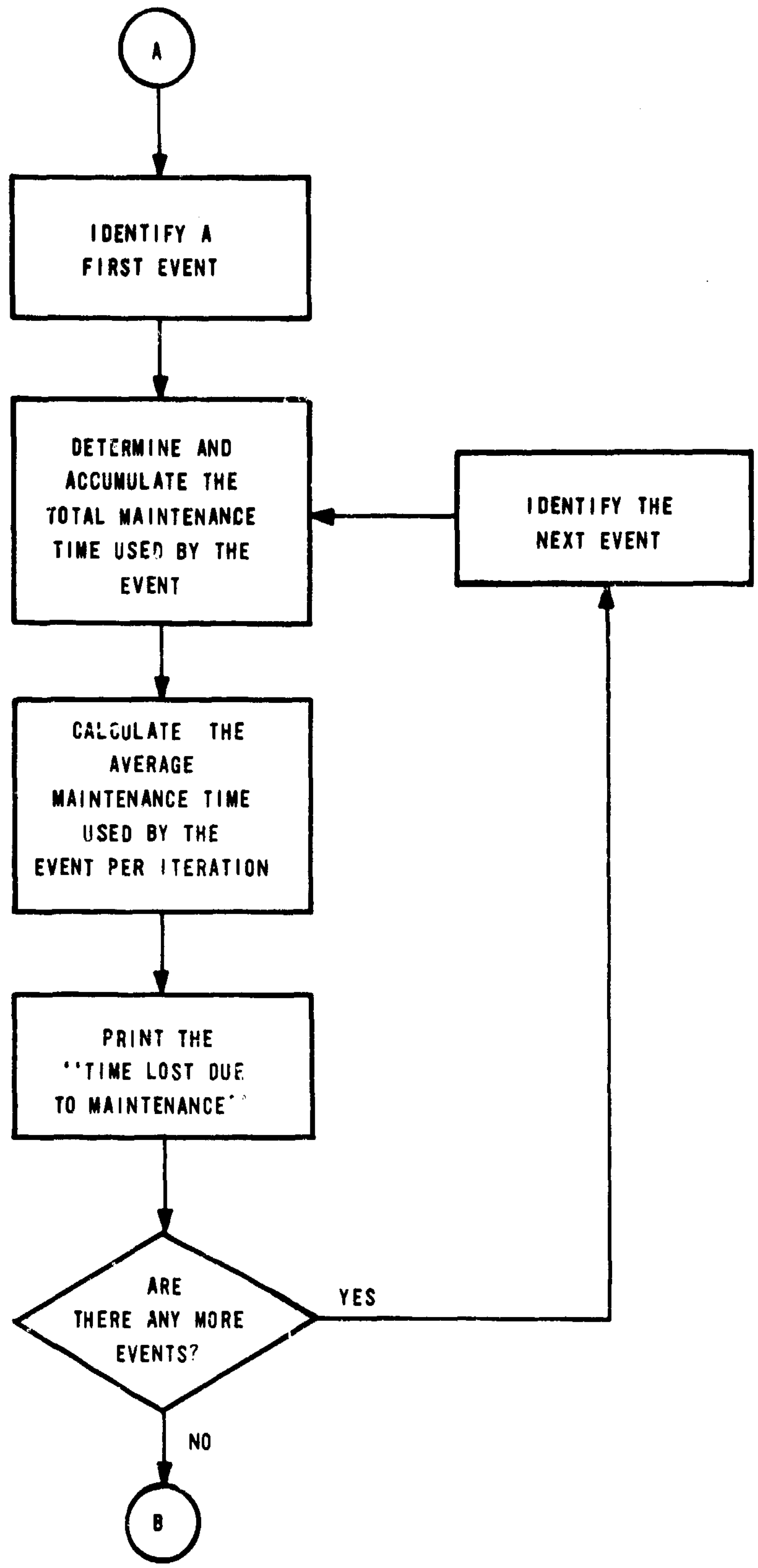


FIG. 19 SUMMARY LOGIC (CONT'D)

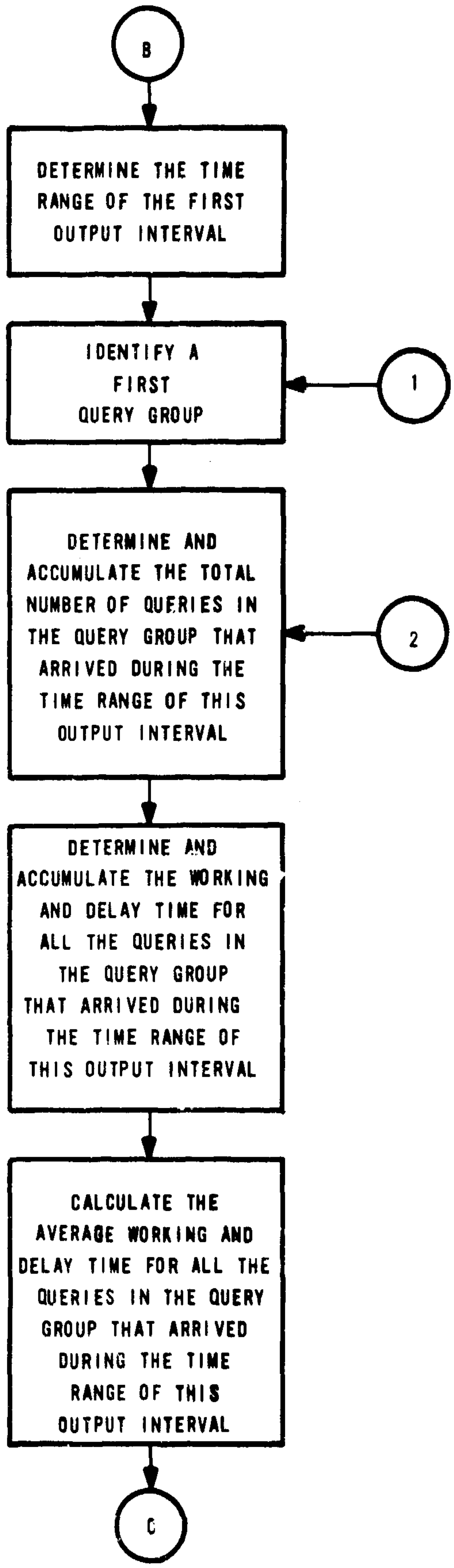


FIG. 19 SUMMARY LOGIC (CONT'D)

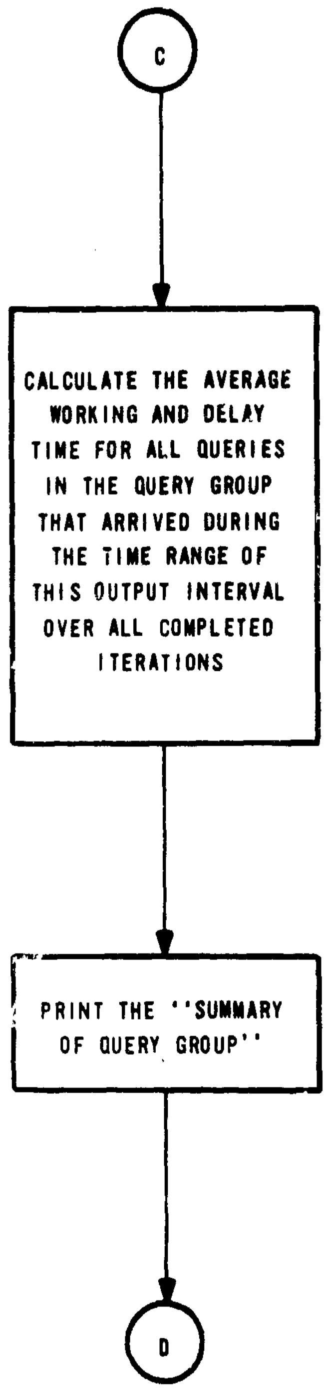


FIG. 19 SUMMARY LOGIC (CONT'D)

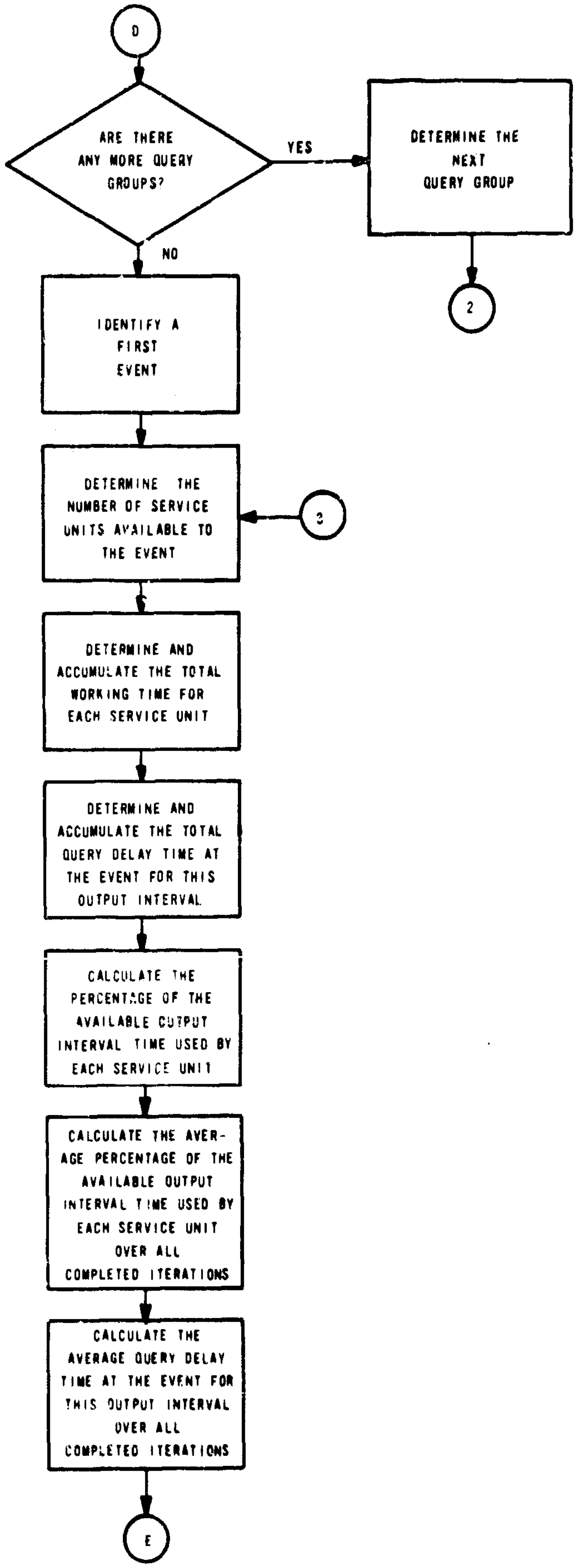


FIG 19 SUMMARY LOGIC (CONT'D)

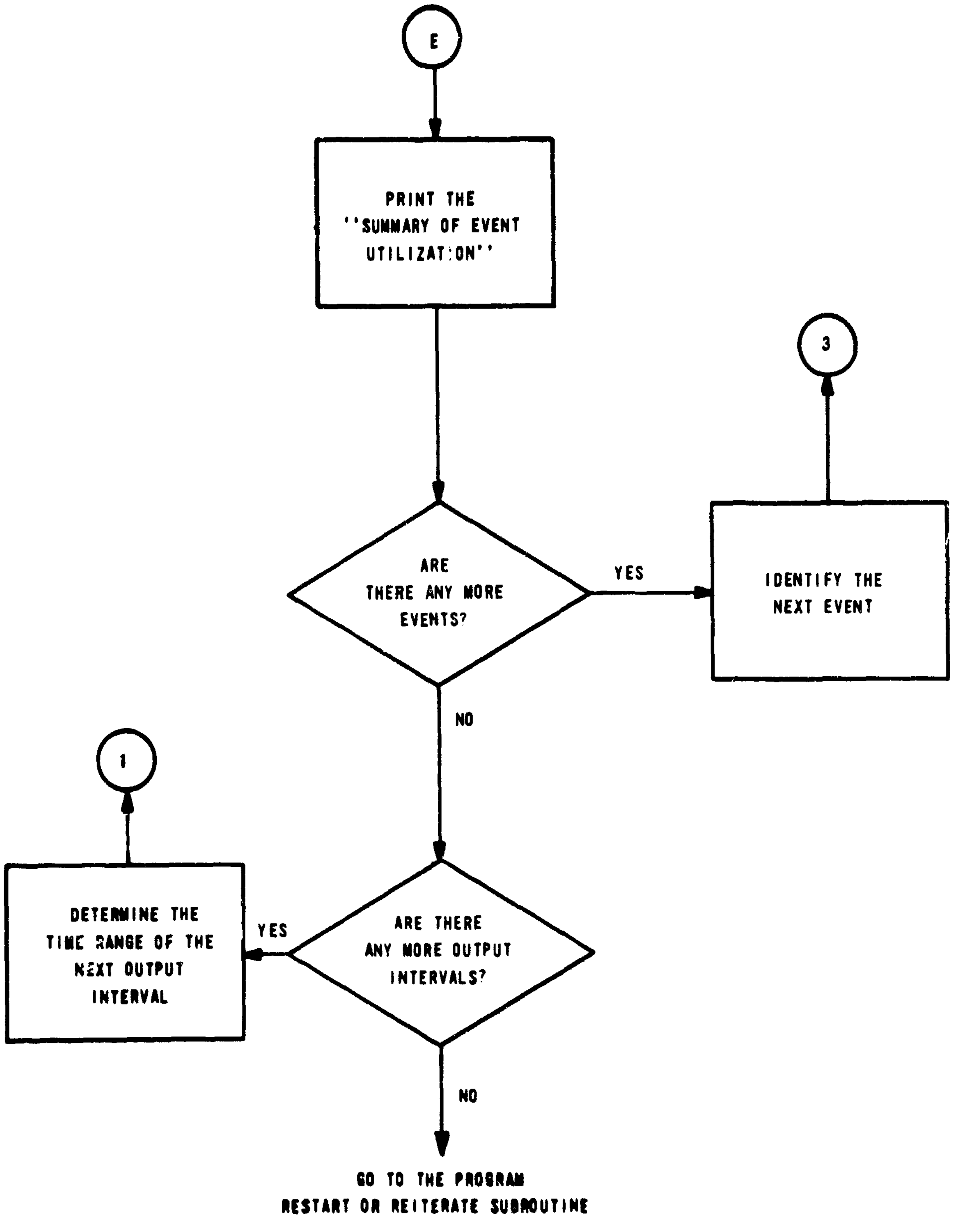


FIG. 19 SUMMARY LOGIC (CONT'D)

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) HRB-Singer, Inc. State College, Pennsylvania	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A GENERAL MODEL FOR SIMULATING INFORMATION STORAGE AND RETRIEVAL SYSTEMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)

Blunt, Charles R., Duquet, Robert T., and Luckie, Peter T.

6. REPORT DATE April 1966	7a. TOTAL NO. OF PAGES 184	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. Nonr 3813(00) b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 352.14-R-2 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. AVAILABILITY/LIMITATION NOTICES

None

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Information Systems Branch Office of Naval Research

13. ABSTRACT

This report presents the results of a research effort to explore the use of computer simulation as a quantitative tool for planning, analyzing and evaluating Information Retrieval (IR) systems. A general time-flow model has been developed that enables a systems engineer to simulate the interactions among personnel, equipment and data at each step in an information processing effort. The input parameters for the simulation reflect the configuration of the system, the processing load of the system, the work schedule of the system, the work schedule of the personnel, equipment availability, the likelihood and effect of errors in processing and the location and availability of the system user. Simulation output provides a study of system response time (both delay time and processing time), equipment and personnel work and idle time and the location and size of data queues.

Included within this report is a discussion of the simulation rationale, the modeling methodology employed and the input and output data of the computer simulation programs. Additionally, one example of a system simulation is presented as an illustration of the capability of this kind of tool in systems analysis.

Unclassified

Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Information Retrieval, Simulation						
Data Storage Systems						
Models						
Evaluation						
Effectiveness						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

Unclassified

Security Classification